AD-A036 437 OFFICE OF TELECOMMUNICATIONS BOULDER COLO INST FOR TE--ETC F/G 17/2
A SIMULATION MODEL FOR ANALYZING PERFORMANCE OF SOME SIMPLE COM--ETC(U)
NOV 76 H AKIMA, A D SPAULDING DOT-FA74WAI-431

UNCLASSIFIED FAA/RD-76/181 NL

1 OF 1
AD-A
036 437

END
DATE
FILMED
4-20-77
NTIS

U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A036 437

A SIMULATION MODEL FOR ANALYZING PERFORMANCE OF SOME SIMPLE COMMUNICATION SYSTEMS ON A DIGITAL COMPUTER

OFFICE OF TELECOMMUNICATIONS
BOULDER, COLORADO

NOVEMBER 1976

Report No. FAA-RD-76-181

ADA036437

A SIMULATION MODEL FOR ANALYZING PERFORMANCE OF SOME SIMPLE COMMUNICATION SYSTEMS ON A DIGITAL COMPUTER

Hiroshi Akima

Arthur D. Spaulding

U.S. Department of Commerce
Office of Telecommunications
Institute for Telecommunication Sciences
Boulder, Colorado 80302

November 1976

Document is available to the U.S. public through
the National Technical Information Service,
Springfield, Virginia 22161.

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D.C. 20590

NOTICE

Technical Report Documentation Page

1. Report No.: FAA-RD-76-181

2. Government Accession No.:

3. Recipient's Catalog No.:

4. Title and Subtitle: A Simulation Model for Analyzing Performance of Some Simple Communication Systems on a Digital Computer

5. Report Date: November 1976

6. Performing Organization Code:

7. Author(s): Hiroshi Akima and Arthur D. Spaulding

8. Performing Organization Report No.:

9. Performing Organization Name and Address:
U.S. Department of Commerce
Office of Telecommunications
Institute for Telecommunication Sciences
Boulder, Colorado 80302

10. Work Unit No. (TRAIS):

11. Contract or Grant No.: DOT-FA74WAI-431

12. Sponsoring Agency Name and Address:
U.S. Department of Transportation
Federal Aviation Administration
Systems Research and Development Service
Washington, D.C. 20590

13. Type of Report and Period Covered:

14. Sponsoring Agency Code: ARD-60

15. Supplementary Notes:

16. Abstract

A digital-computer simulation model for analyzing performances of some simple communication systems is being developed for the purpose of studying the effects of interfering signals, noise, and/or distortions on various communication systems. Development of the model is largely based on the analogy with the laboratory tests of communication system performances. The model consists of computer subprograms, each of which either simulates a basic component of communication systems or calculates characteristics of a component. An introductory explanation of computer simulation, guidelines for developing a simulation model, an outline of the model, and some example programs are presented. Complete Fortran listings and a user manual are also presented.

17. Key Words: Communication system, communication system performance, computer simulation, digital computer, distortion, interference, noise.

18. Distribution Statement: Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151

19. Security Classif. (of this report): Unclassified

20. Security Classif. (of this page): Unclassified iii

21. No. of Pages: 85

22. Price:

Form DOT F 1700.7 (8-72)

METRIC CONVERSION FACTORS

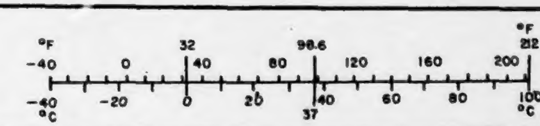
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in ≡ 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



FEDERAL AVIATION ADMINISTRATION
SYSTEMS RESEARCH AND DEVELOPMENT SERVICE
SPECTRUM MANAGEMENT STAFF

STATEMENT OF MISSION

The mission of the Spectrum Management Staff is to assist the Department of State, Office of Telecommunications Policy, and the Federal Communications Commission in assuring the FAA's and the nation's aviation interests with sufficient protected electromagnetic telecommunications resources throughout the world to provide for the safe conduct of aeronautical flight by fostering effective and efficient use of a natural resource — the electromagnetic radio frequency spectrum.

This objective is achieved through the following services:

- Planning and defending the acquisition and retention of sufficient radio frequency spectrum to support the aeronautical interests of the nation, at home and abroad, and spectrum standardization for the world's aviation community.

- Providing research, analysis, engineering, and evaluation in the development of spectrum related policy, planning, standards, criteria, measurement equipment, and measurement techniques.

- Conducting electromagnetic compatibility analyses to determine intra/inter-system viability and design parameters, to assure certification of adequate spectrum to support system operational use and projected growth patterns, to defend aeronautical services spectrum from encroachment by others, and to provide for the efficient use of the aeronautical spectrum.

- Developing automated frequency selection computer programs/routines to provide frequency planning, frequency assignment, and spectrum analysis capabilities in the spectrum supporting the National Airspace System.

- Providing spectrum management consultation, assistance, and guidance to all aviation interests, users, and providers of equipment and services, both national and international.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

A SIMULATION MODEL FOR ANALYZING PERFORMANCE OF SOME SIMPLE COMMUNICATION SYSTEMS ON A DIGITAL COMPUTER

Hiroshi Akima and Arthur D. Spaulding*

A digital-computer simulation model for analyzing performances of some simple communication systems is being developed for the purpose of studying the effects of interfering signals, noise, and/or distortions on various communication systems. Development of the model is largely based on the analogy with the laboratory tests of communication system performances. The model consists of computer subprograms, each of which either simulates a basic component of communication systems or calculates characteristics of a component. An introductory explanation of computer simulation, guidelines for developing a simulation model, an outline of the model, and some example programs are presented. Complete Fortran listings and a user manual are also presented.

Key Words and Phrases: Communication system, communication system performance, computer simulation, digital computer, distortion, interference, noise.

1. INTRODUCTION

For efficient use of radio-frequency spectrum it is important to know how communication systems perform when desired signals are received with interfering signals and/or noise or suffer from distortions due to excessive band limiting or other causes. Analyzing system performances in such cases is not always easy, partly because demodulators in most systems are nonlinear circuits. One way of supplementing our knowledge of system performances is through simulation of communication systems on a digital computer. The Institute for Telecommunication Sciences of the Office of Telecommunications (OT/ITS) is developing a computer simulation model for this purpose. This report describes the efforts for developing the model and the characteristics of the resulting model as well as necessary information for the use of the model.

The idea of computer simulation of communication systems is not new. As an example, we can mention a computer simulation model described in CCIR Report 520 (CCIR, 1975). This CCIR Report, however, does not include program listings of the model. To the authors' knowledge, no program listings of simulation models have been published in the open literature.

According to Webster's dictionary, the verb "simulate" means "to have the external characteristics of" or "to act like". Therefore, computer

*The authors are with the Institute for Telecommunication Sciences, Office of Telecommunications, U.S. Department of Commerce, Boulder, Colorado 80302.

simulation of a communication system is to let the computer have the external characteristics of or act like a communication system. The input data to be given to the computer can be in several forms — time-sampled voltage values of the input signal, frequency-spectrum components of the input signal, system parameters for the modulator, etc. In any case, one must have instructed the computer, through programming in advance, how to act like the communication system on given data. The output data from the computer can also take several forms — time-sampled voltage values of the output signal, frequency-spectrum components, signal-to-interference ratio (SIR) or signal-to-noise ratio (SNR) values, etc. The simulation model is a major part of the instruction to be given to the computer, and a well-designed model greatly reduces the user's difficulty in performing computer simulation. (The reason that the simulation model cannot be the complete instruction will be explained later.)

In this report, an introductory explanation of computer simulation of communication system performance is given in section 2; the parallel relationship between the computer simulation and the laboratory tests is emphasized. In section 3, guidelines for developing the model are derived. In section 4, a general description of the model developed in this study is given. Section 5 describes how to use the model; some simple examples are included to illustrate its use. Section 6 summarizes all preceding sections and gives some suggestions for further development. Fortran listings of all subprograms included in the model are given in Appendix A, and a user manual is given in Appendix B.

2. SIMULATION OF SYSTEM PERFORMANCE ON A DIGITAL COMPUTER

2.1. Need for Computer Simulation

Typically, there are three methods of studying performances of communication systems including the performance of demodulators; i.e., theoretical analysis, laboratory test, and computer simulation. Each has its advantages. Theoretical analysis is convincing if the analysis is done without approximation errors. There are, however, many cases in which theoretical analyses do not look feasible even if not altogether impossible. A laboratory test is also good if the test is done in a well-controlled situation. In many cases, however, laboratory tests lack flexibility; changing a system parameter sometimes necessitates rebuilding of all of the equipment in the test arrangement. Laboratory tests can also be quite expensive and time consuming, especially tests for designing new systems. Computer simulation can mitigate the disadvantages of the other two methods: it is almost always possible (at least theoretically if sufficient time is allowed for computation) and it is flexible.

As a typical example we will take a frequency-modulation (FM) demodulator and first consider the case where an FM desired signal is received with white Gaussian noise. The output signal-to-noise ratio (SNR) has been previously calculated as a function of input SNR and the modulation index (i.e., the ratio of peak frequency deviation to maximum frequency of the baseband signal) (Stumpers, 1948; Akima, 1963). Second, we consider the case where the desired signal is received with non-Gaussian noise. We can even consider

the third case where the desired signal is received with an interfering signal and noise (either Gaussian or non-Gaussian). Very little is known about the output SNR for these cases. Solving even these simple problems (e.g., calculating the output SNR) by theoretical analysis is too complicated.

We may ask: "Can we solve the same problems with laboratory tests?" The answer is "Yes," of course, but we need some further considerations. As mentioned earlier, the test set-ups are not simple even for these simple problems. If we design all the components such as the modulator, noise generator, bandpass filter, and demodulator to be flexible enough for possible changes in system parameters, the cost will be very high. If we disregard the significance of flexibility in initial test set-ups, changing the modulation index, for example, may necessitate rebuilding of the modulator, bandpass filter, and the demodulator.

This is a good example of one way in which computer simulation can do a better job. As will be explained later, we can write several computer subprograms that simulate system components such as the FM modulator, bandpass filter, FM demodulator, etc. If properly written, a computer subprogram is flexible in nature. A subprogram that simulates an FM modulator, for example, works for any legitimate values of system parameters such as the baseband bandwidth, center frequency, modulation index, etc. Writing a flexible subprogram does not usually require extra effort or extra cost, both of which are required for building a flexible hardware system component. Running laboratory tests with various sets of parameter values is equivalent to calling the subprograms with various sets of parameter values. Computer simulation is very useful when system performance is studied for system parameters of wide variability.

In the laboratory tests, care must be taken so that test conditions will not change inadvertently. For example, stabilizing the power supply voltage is required in many cases. Room temperature can be another unstable condition. All parameter values except those which one wants to change intentionally must be kept constant. Computer simulation takes care of this requirement automatically. In computer simulation, only those parameters one wants to change are changed, and other parameters remain constant.

When one studies the effects of an interfering signal of a specified type, for example, all test equipment must be free from other unintentional interferences and/or noise (either electromagnetic, acoustic, or mechanical). In other words, intensities of unintentional interference and noise must be negligibly small compared with those of the desired signal and of the interfering signal in question. Computer simulation can take care of this, too, very easily. The main source of "noise" in computer simulation is the one due to errors in binary approximation of real numbers, but it is usually very small. If one simulates, for example, a back-to-back test of an FM system in which an FM demodulator is connected to an FM modulator directly, an output SNR of 200 dB or higher is obtained on a CDC-6000 series computer.

Computer simulation was first considered as a supplement to laboratory tests in the beginning of this section, but the last two paragraphs indicate that computer simulation can do more than laboratory tests in certain aspects. One can simulate an idealized laboratory test very easily.

2.2. Computer Simulation of System Performance

Computer simulation of a communication system allows the computer to act like a communication system or like a part of the system. Assuming a typical example, we will first describe what will take place in the computer simulation.

Suppose one is studying the effects of an AM interfering signal on an FM desired signal in the presence of white Gaussian noise. The computer produces data (a sequence of numbers) that represent the FM desired signal with specified system parameters such as the frequency of the modulating signal, the modulation index, and the center frequency of the modulated signal. The data can be a sequence of time-sampled voltage values or a frequency spectrum (i.e., Fourier transform of the time sequence). (The choice as to whether a time sequence or its spectrum is used is a matter of technicality; it is discussed in detail later.) Similarly, the computer generates data for the AM interfering signal and data for the noise. Next, the computer superposes these three sets of data with relative ratios that correspond to a specified signal-to-interference ratio (SIR) and a signal-to-noise ratio (SNR); the resulting data represent the input signal to the receiver. Next, the computer acts like an intermediate-frequency (IF) bandpass filter; it modifies the receiver input signal and obtains the filter output signal. Then, the computer acts like an FM demodulator; it produces a baseband signal that corresponds to the demodulator output. If so instructed, the computer does further baseband processing; it acts like a baseband filter or a psophometer. (A psophometer is an apparatus for the objective measurement of baseband noise, it is essentially a bandpass filter that represents average frequency response of human ears and the frequency response of a typical electro-acoustic transducer in the receiver. The psophometrically-weighted noise power is usually a good measure of the disturbing effect of the baseband noise.) The computer can even act as a monitoring device such as an oscilloscope or a spectrum analyzer; it can display a signal at any point in the system as a time sequence or a spectrum.

As described in this example, the computer acts like each system component along the flow of the signal in the system. (To simulate two system components or more that operate in parallel such as the FM and AM modulators and the noise generator in this example, the computer simulates one component at a time in an arbitrary order.) The flow of data in the computer simulation is shown schematically in figure 1. It is clear from this figure (and also from the preceding descriptions) that signal flow in a computer simulation parallels the flow in a laboratory test. This parallel relationship between computer simulation and laboratory tests is very important in developing the model for computer simulation as well as in using the model for solving specific problems.

In the development of a computer program for system simulation, it is advisable to set apart the portions of the program that correspond to isolated procedures. The more subprograms one can write, the simpler the main program will be. Since it is almost impossible to please every user in the way of writing the input and output parts of the program, it is desirable that each procedure that simulates each system component — each box in figure 1 — be written as a subprogram, leaving it up to the user to determine the main program which makes use of the appropriate subprograms.

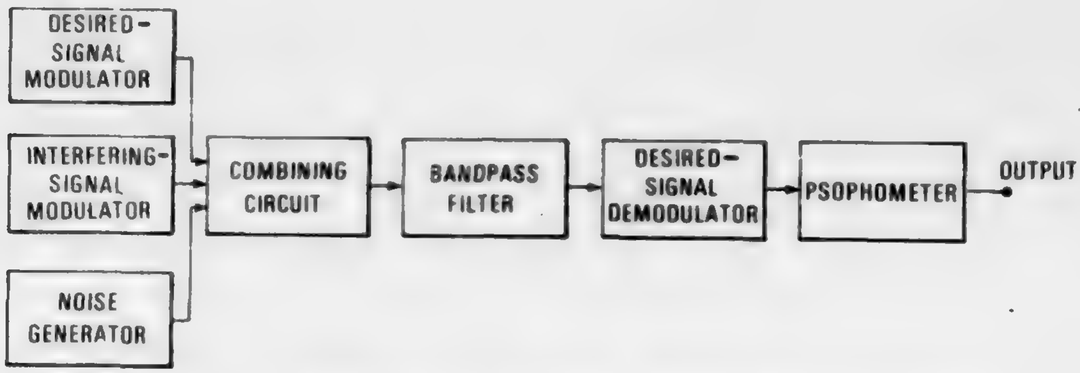


Figure 1. An example of computer simulation arrangement. (A voice signal case.)

3. GUIDELINES FOR DEVELOPING A SIMULATION MODEL

3.1. Main Structure of Computer Simulation Model

In the preceding section, we have considered the parallel relationship between a computer simulation model and laboratory test facilities and we have suggested that a computer simulation model of communication system performance will be a collection of computer subprograms, each simulating a system component. In this section, we proceed with the parallel relationship and determine the main structure of the computer simulation model.

Suppose a communication engineer is studying in the laboratory the performance of a communication system in which the desired signal is received in the presence of an interfering signal and noise. The first thing he does when he enters the laboratory is perhaps to locate necessary system components such as the modulator, demodulator, noise generator, etc., on the shelves of the laboratory. If he can locate necessary components, he brings them to the test bench. He must buy, borrow, or build the missing components, if any. Next, he makes necessary cable connections among these components. To do this and also to conduct the test that follows, he usually needs a user manual for each component. Before starting the test, he must consider and decide many things. The ranges and increments of the signal-to-interference ratio (SIR) and signal-to-noise ratio (SNR), the modulation indexes, the frequency detuning of the signals are only a few of such items. Decisions on these items is his responsibility, not of the laboratory facilities. The facilities usually include various items of test equipment such as voltmeters, power meters, VU (volume unit) meters for voice signals, CRT (cathode-ray-tube) oscilloscopes, etc., and the selection of these pieces of equipment for a specific test is also his responsibility. During the test, he may have to modify his test plan, depending on the partial results of the test. Finally, he tries to present the test results in a comprehensive manner. The decision on the means of presentation (i.e., tables, graphs, empirical formulas, etc.) or on the format of the tables or graphs is also his responsibility. This example suggests several guidelines for developing laboratory test facilities for communication system performances.

The above example indicates clear distinction between the responsibility of the laboratory and that of the user. The laboratory including its maintenance staff, if any, does not conduct a test by itself. All details of planning and conducting the test and of representing the test results are the responsibility of the user. A good laboratory is the one that provides the maximum ease to the user. Since the computer simulation model is a software replica of a test laboratory, the model does not perform simulation by itself; the user must do this by writing a computer program. A good computer simulation model is the one that keeps the difficulties to the user to a minimum.

A good laboratory must have a variety of communication system components. Their types and operating ranges must cover the users' needs. Each component must work properly. It must be well documented. Standardized cable connectors and power supplies are very desirable for easing the test-arrangement efforts. These are some of the guidelines for developing good laboratory test facilities.

Translation of these guidelines for good laboratory test facilities to those for a good computer simulation model is almost verbatim. A good computer simulation model must have a variety of computer subprograms, each simulating a communication system component. Subprogram types and operating ranges must cover the users' needs. (In general, however, satisfying the requirements for wide operating ranges of parameters for a computer subprogram is not as severe as for a hardware system component.) Each subprogram must do exactly what it is expected to do; it must have been completely debugged. Each subprogram must be well documented so that every user can call it without difficulty. Procedures of calling these subprograms must be standardized and consistent within the model. A model must be written in a standardized computer language such as American National Standards Institute (ANSI) Standard Fortran (ANSI, 1966) so that the model can be used on most computers.

3.2. Frequency-Domain vs. Time-Domain Approach

The analogy with the laboratory test has been extensively used above. But there are several things to which this analogy does not apply. The subject of this section, i.e., the discussion of frequency-domain approach vs. time-domain approach, is one of them.

In laboratory tests, a signal voltage is continuous most of the time. Because of the sampling theorem (Wozencraft and Jacobs, 1967, ch. 8), a band-limited signal can be represented by a sequence of signal voltage values that are measured at sampling points equally spaced in time. (The sampling rate must be equal to or higher than the so-called Nyquist rate that is equal to twice the maximum frequency of the signal.) Representation of a signal by a sequence of time-sampled voltage values is not the only way of representing a signal. It can also be represented by its Fourier transform, or a sequence of frequency spectrum components. These two representations are equivalent mathematically, and transformation from one to the other can be done easily on the computer.

Since these two representations are mathematically equivalent, the choice of one depends on users' convenience. Problems associated with this choice are discussed below. Let a filter (either a bandpass or a lowpass filter) be a first example. In general, characteristics of a filter are commonly expressed by its frequency responses; i.e., amplitude response (or attenuation) and phase response (phase delay) versus frequency. Filter characteristics can also be expressed by the impulse response, i.e., the transient output waveform when a unit impulse is imposed at the filter input. Effects of a filter can be calculated either by multiplying, on a component-by-component basis, the spectrum components of the input signal by the filter frequency responses in frequency domain, or by convolving the input signal with the impulse response in time domain. These two methods of calculation are, again, equivalent mathematically, but they are very different in degree of convenience of numerical calculation. Frequency response of a filter substantially diminishes if the frequency deviates from the filter passband to some extent, but it takes a long time for the impulse response to diminish substantially. In the frequency domain, a spectrum component of the output signal is a simple product of the respective spectrum component of the input signal and the frequency response of the filter, but in the time domain, each sampled value of the output signal is an integral (usually a truncation of an infinite integral) of the product of the input signal and the filter impulse response. Therefore, we can see that the frequency-domain representation is much more convenient for representing the characteristics of a filter.

Modulators will be considered next. To generate a double-sideband (DSB) amplitude-modulation (AM) signal, either time of frequency domain approach is equally simple and straightforward. For a single-sideband (SSB) signal, the frequency-domain approach is much simpler than the time-domain approach; SSB modulation is equivalent to shifting the spectrum components of the baseband signal (modulating signal) to the radio frequency region. For other types of modulations such as phase modulations or frequency modulations (FM), generating a signal as a function of time is much simpler than generating its spectrum directly.

As for the demodulators, the situation is a little different except for the SSB demodulator, for which the frequency-domain approach is simpler. To demodulate a DSB-AM signal, the time-domain approach is simpler. For an FM signal, a method starting with the spectrum is not only simpler but also more accurate, as described later.

From these observations, we can conclude that there is no clear basis for selecting one over the other, insofar as the modulators and demodulators are concerned. In this situation, convenient representation of filter characteristics with the frequency response might be a decisive factor. Of course, it is possible to use one representation for some system components and the other representation for other components, but we reject this idea. Such an idea would confuse the prospective users of the model; it is not easy for many users to remember which representation is used for the input and output of a specific system component.

3.3. Lower-Frequency Approach

Complexity of computation increases with the increase in number of sampling points (or number of frequency components) to be handled by the model. It is desirable to keep these numbers as small as possible. To reduce the numbers, let us review a simple rule that governs the number of sampling points (or the number of frequency components).

Suppose we wish to represent a voltage waveform of duration T, which is a real function of time t, and we do this by taking N voltage values at N sampling points equally spaced in time. The minimum number N that is necessary to represent the exact waveform depends on the fineness of the waveform or the maximum frequency component contained in the waveform. According to the sampling theorem, the sampled values taken at the rate of 2 f_m can exactly represent the waveform when the waveform does not contain any frequency spectrum components higher than f_m. Therefore, if we take

$$N' = 2 f_m T$$

samples equally spaced in time for a waveform having a duration T and maximum frequency f_m, these sampled values exactly represent the waveform. The same number applies to the number of frequency spectrum components that represent the same waveform. As is well-known, such a waveform can be represented with a series of sine and cosine functions having a fundamental frequency 1/T and harmonic frequencies i/T, where i = 2,3,... . Since no component exists that is higher than f_m, there are $N' = 2 f_m T$ components in the series; i.e., $f_m T$ sine components and $f_m T$ cosine components.

In the preceding paragraph, we have represented everything with real numbers. From now on, however, we will use complex numbers. We represent the waveform with a complex function of a real argument t; the actual waveform is a real part of this complex function. The cosine and sine functions of an identical angle are combined into a complex exponential function as

$$\exp (jx) = \cos x + j \sin x.$$

Using complex-number representations, we can represent a waveform of duration T and of maximum frequency f_m with either

$$N = f_m T$$

time sampled values or N frequency spectrum components.

It is clear from the above considerations that reducing N is equivalent to reducing the frequency-time produce, $f_m T$. A simple way of achieving this is to lower f_m as much as possible. Let us consider an example. The spectrum of an AM signal modulated with a voice signal having the maximum frequency of 3 kHz extends from 3 kHz below the carrier frequency to 3 kHz above the carrier frequency. If the carrier frequency is 100 kHz, f_m is 103 kHz. If we consider the signal in an interval of 10 ms, N must be at least 1030. But, if we lower the carrier frequency to 5 kHz, f_m is 8 kHz, and N for 10 ms is 80. In the laboratory tests, modulating a carrier of 5 kHz with a

modulating signal of 3 kHz is neither easy nor advantageous, but is is definitely good in computer simulation.

This lower-frequency approach should not be confused with the so-called lowpass-equivalent analysis that is used for the analysis or synthesis of a bandpass filter. Our approach is equivalent to beating down the frequency (similar to the process used in a superheterodyne receiver).

3.4. Basic Parameters for Simulation Model

One of the most basic parameters in the computer simulation model is the number of frequency-spectrum components (or the number of sampling points). This number dictates the necessary core-memory size in the computer and determines the computation time required for simulation. Since this number is the product of the maximum frequency of the signal and the time duration of the signal, the latter two are also important parameters.

The frequency of the signal in the computer simulation model does not have to be equal to the frequency of the signal to be simulated. As in the laboratory tests, one can choose an arbitrary frequency as the center frequency of the signal, as long as the relative relations among the center frequencies of the signals and the bandpass filter are kept unchanged and the minimum frequencies of all the signals are above zero. The maximum frequency used in the model must be equal to or greater than the maximum frequencies of all the signals involved.

Representing a waveform of duration T with frequency spectrum components implies that we are considering a waveform of infinite duration consisting of infinite repetitions of the waveform of duration T. Even if the model uses the time-domain approach and represents the input and output voltage waveforms with sequences of time-sampled voltage values, use of frequency spectra of these waveforms cannot be avoided. Rigorously speaking, therefore, computer simulation of the type we are considering can handle a waveform only when the waveform is a periodic function of T.

If we take the frequency-domain approach, the fundamental frequency or the unit frequency spacing of the spectrum, f_0, is an important parameter. It is the reciprocal of T and is equal to f_m/N. If we take the time-domain approach, the unit time interval of sampling, t_0, is an important parameter. It is the reciprocal of f_m and is equal to T/N.

So far we have identified five parameters; i.e., the number of frequency spectrum components (or the number of sampling points) N, the unit frequency spacing in the spectrum f_0, the maximum frequency f_m, the unit time interval of sampling t_0, and the time duration of the signal T. These five parameters have their own physical meanings, but they are not independent of each other. It is very easy to show that, if we select a pair of independent parameters, the remaining three parameters can be represented as simple functions of the former two. We will select two parameters for the model in the paragraph that follows.

Since N is the basic parameter in computer programming, we will select N as a basic parameter for the model. For representing the characteristics of modulators and demodulators, any one out of the remaining four parameters can be used as the basic parameter. For representing the characteristics of a filter, however, neither t_0 nor T has a good and comprehensive meaning. As for the choice of one out of f_0 and f_m, f_0 is more convenient for representing other frequencies such as the center frequencies of signals or the bandwidths of filters. In summary, we will select the number of frequency spectrum components N and the unit frequency spacing in the spectrum f_0 as basic parameters for the computer simulation model.

3.5. Programming Language

In general, a Fortran language is accepted by most computer system installations today. A Fortran language is one of the basic computer languages taught in beginning programming classes. Learning Fortran programming does not take a long time for average scientists and engineers. Perhaps the only tedious, but not difficult, part of Fortran programming is specifying the proper format of data in conjunction with the input and output statements.

There are several versions (or dialects) of Fortran that particular computer systems can accept. Conversion from one version of Fortran to another is sometimes difficult. For the purpose of minimizing the conversion problems to a specific version, ANSI standard Fortran (ANSI, 1966) has been developed. Most Fortran compilers can compile programs that are written in ANSI standard Fortran. When revisions of programs written in ANSI standard Fortran are required by a computer system, such revisions are usually minor. In such a case, the user manual of the computer system usually gives instructions on how to revise the programs, or systems people can help the user modify his program. Therefore, the model is written in ANSI standard Fortran.

4. GENERAL DESCRIPTION OF THE MODEL

4.1. Outline of the Model

The computer simulation model developed here is a collection of computer subprograms, each of which, except the one for Fourier transforms, either simulates a basic component of communication systems or calculates characteristics of a component. The model does not include a main program; a user must write his own main program, usually in the Fortran language, to perform computer simulations.

The model includes 19 subprograms; i.e., five for modulators, five for demodulators, two for signal generators, two for a bandpass filter, two for psophometers, two for radio-frequency (RF) combining circuits, and one for discrete Fourier transforms. These 19 subprograms are listed in table 1 with respective brief descriptions of action. Complete Fortran listings of the subprograms are given in Appendix A, and their user write-ups are given in Appendix B.

Table 1. List of Subroutines Included in the Model

Name	Brief Description of Action
BPFCH	Simulates a bandpass filter of Chebyshev type.
CBCCT2	Simulates an RF combining circuit that linearly combines two RF signals.
CBCCT3	Simulates an RF combining circuit that linearly combines three RF signals.
DEMAM	Simulates an AM demodulator.
DEMFM	Simulates an FM demodulator.
DEMFSK	Simulates an FSK demodulator.
DEMPHM	Simulates a phase-modulation demodulator.
DEMSSB	Simulates an SSB demodulator.
DFT	Performs discrete Fourier transform.
FRBPFC	Calculates frequency response of a bandpass filter of Chebyshev type.
MODAM	Simulates an AM modulator.
MODFM	Simulates an FM modulator.
MODFSK	Simulates an FSK modulator.
MODPHM	Simulates a phase-modulation modulator.
MODSSB	Simulates an SSB modulator.
PSPHCT	Simulates a psophometer for commercial-telephone circuit.
PSPHPT	Simulates a psophometer for program-transmission circuit.
SGPLS	Simulates a pulse-signal generator.
SGWGN	Simulates a white-Gaussian-noise generator.

All subprograms included in the model are written in ANSI standard Fortran (ANSI, 1966). They can be called by a Fortran program.

All signals and noise are represented as their complex voltage spectra at the input and output of each subroutine. This means that, insofar as the user is concerned, the frequency-domain approach is adopted, although signals are transformed to time functions in some subroutines. The spectrum of each signal or noise is a complex voltage spectrum.

The power spectral density of the output of the Gaussian noise generator subroutine is unity. The output powers of all signal generator and modulator subroutines are normalized in such a way that the output power is equal to unity, where the ITU Radio Regulations (ITU, 1968) means for specifying the power of an emission are used.

Because of the use of a simplified fast Fourier transform (Cooley and Tukey, 1965; Cochran et al., 1967), the number of complex spectrum components of the input and output signals must be a power of 2. In addition, we have imposed a restriction that the number of complex spectrum components must not exceed 2^{14}.

Some subroutines included in the model call the RANF(X) function, which generates a pseudo-random number uniformly distributed in the interval between 0 and 1. (The parameter X is a dummy parameter.) When a user's computer system does not include the RANF function or its equivalent as a library function, the user must supply the function.

A prospective user of this computer simulation model must have a good knowledge of communication systems; otherwise he cannot perform meaningful simulation. By a good knowledge of communication systems, we do not mean details of communication systems. What is meant by a good knowledge here is a good common-sense knowledge regarding the power of emission (or transmitter power), noise power, noise density, signal-to-interference ratio, signal-to-noise ratio, received signal quality, etc., as well as good understanding of meanings of system parameters for the systems he plans to simulate.

A prospective user of this model must know how to use a computer and must have a good knowledge of Fortran, especially the ANSI standard version of Fortran. This, however, does not mean that a prospective user must be an expert in computer programming.

The size of the computer required for simulation with this model depends on the size of simulation, i.e., the number of frequency components (or equivalently, the number of sampling points) for analog signals and the length of the bit sequence in digital systems. In many applications, however, the required computer size is not a severe problem. Since the computer must have a Fortran compiler which requires a large core size, we can generally say that a computer can run a program with this model if it can compile the model.

4.2. Mathematical Bases

Mathematical bases for some subprograms included in the model are described in this section. Modulators and signal and noise generators are described first, followed by the descriptions of demodulators, bandpass filter characteristics, and psophometers.

AM and SSB Modulator: These subroutines generate, from the spectrum components of the modulating signal, the sideband components in the frequency domain.

FM Modulator: This subroutine integrates the modulating signal with respect to time and phase modulates the carrier with the integrated modulating signal in the same manner as in the phase-modulation modulator (described below). Integration with respect to time is equivalently performed in the frequency domain by dividing each frequency component of the signal by $j\omega = j2\pi f$.

FSK Modulator: This subroutine generates an FSK signal in the time domain and Fourier transforms the signal to the frequency domain.

Phase-Modulation Modulator: This subroutine Fourier transforms the modulating signal from the frequency domain to the time domain, phase modulates the carrier with the transformed signal in the time domain, and Fourier transforms the modulated signal to the frequency domain.

Pulse-Signal Generator: This subroutine generates a pulse signal in the time domain and Fourier transforms the generated signal to the frequency domain.

White-Gaussian-Noise Generator: This subroutine generates, for each complex component of the noise, a uniformly distributed random number, converts the number to a Rayleigh distributed random number, and multiplies the number by cosine and sine of an arbitrary (uniformly distributed) phase angle to obtain the real and imaginary parts of the spectrum component. This procedure is based on the method suggested by Box and Muller (1958).

AM and Phase-Modulation Demodulators: These subroutines Fourier transform the input signal from the frequency domain to the time domain, extract the amplitude (or envelope voltage) and the phase of the signal, respectively, in the time domain, and Fourier transform the extracted information from the time domain to the frequency domain.

FM Demodulator: This subroutine calculates the instantaneous frequency of the input signal from the spectrum of the input signal and Fourier transforms the instantaneous frequency from the time domain to the frequency domain. The instantaneous frequency is calculated as follows:

We assume that the input signal is a narrow-band signal; i.e., the input signal can be expressed as a complex function of time t by

$$e(t) = A(t) \exp[j\theta(t)],$$

where $A(t)$ and $\theta(t)$ are real functions of t representing the instantaneous amplitude and phase, respectively. Taking the natural logarithms of both sides of this equation, we have

$$\ln e(t) = \ln A(t) + j\theta(t).$$

Since $A(t)$ is a real function of t, $\ln A(t)$ is also a real function of t. Therefore, $\theta(t)$ can be expressed by

$$\theta(t) = \mathrm{Im}\{\ln e(t)\},$$

where Im{...} stands for "the imaginary part of." Since the instantaneous frequency $f(t)$ is the time derivative of the instantaneous phase divided by 2π, it is expressed by

$$f(t) = \frac{1}{2\pi}\frac{d\theta(t)}{dt} = \frac{1}{2\pi}\,\mathrm{Im}\{\frac{d}{dt}\ln e(t)\} = \frac{1}{2\pi}\,\mathrm{Im}\{\frac{de(t)}{dt}/e(t)\}.$$

When $e(t)$ is a periodic signal with its fundamental frequency f_0 and is expressed by

$$e(t) = \sum_k C_k \exp[j2\pi k f_0 t]$$

with complex constants C_k, its time derivative can be expressed by

$$\frac{de(t)}{dt} = j2\pi f_0 \sum_k kC_k \exp[j2\pi k f_0 t].$$

From these equations, therefore, we have

$$f(t) = \mathrm{Re}\{f_0 \sum_k kC_k \exp[j2\pi k f_0 t] / \sum_k C_k \exp[j2\pi k f_0 t]\}$$

as an expression for the instantaneous frequency, where Re{...} stands for "the real part of."

FSK Demodulator: This subroutine calculates the instantaneous frequency of the input signal from the spectrum of the input signal in the same manner as in the FM demodulator and estimates the keying bit sequence. Bit synchronization is established so as to maximize the crosscorrelation between the sequence of instantaneous frequency values and the estimated bit sequence.

SSB Demodulator: This subroutine demodulates the input signal by shifting down the spectrum components in the frequency domain.

Calculation of Frequency Response of a Chebyshev-Type Bandpass Filter: It is based on the description given by Storer (1957). Necessary equations are listed as follows:

Frequency response of a Chebyshev-type bandpass filter is uniquely specified by the following parameters:

B = 3-dB bandwidth,
f_c = center frequency,
n = order,
R = ripple in the passband in dB.

The complex frequency response $F(f)$ can be expressed by

$$F(f) = F(f_c) \cdot \prod_{k=1}^{n} \frac{P_k}{\frac{j2r(f-f_c)}{B} - P_k},$$

where $F(f_c)$ = 1 when n is odd,
= $1/V_r$ when n is even,
V_r = ripple in the passband in voltage ratio
= $10^{R/20}$,
P_k = pole of the filter ($k = 1,2,\ldots,n$),
r = scale factor.

The pole P_k on a complex plane is given by

$$P_k = -\sinh u_0 \cos \phi_k + j \cosh u_0 \sin \phi_k,$$

where
$$u_0 = [\sinh^{-1}(1/\varepsilon)]/n,$$
$$\varepsilon = (V_r^2 - 1)^{\frac{1}{2}},$$
$$\phi_k = (2k-n-1)\pi/(2n).$$

The scale factor r is the largest real root of

$$T_n(r) = 1/\varepsilon,$$

where $T_n(x)$ is a Chebyshev polynomial of the first kind of the n^{th} order (Abramowitz and Stegun, 1964). This polynomial is defined by

$$T_n(x) = \cos(n \cos^{-1} x).$$

It can also be calculated from the following recurrence relation

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

with
$$T_0(x) = 1,$$
$$T_1(x) = x.$$

Numerical solution is required for determining the scale factor r for a given value of ε when n equals 3 or greater.

Psophometers: These subroutines are based on the weighting-coefficient curves, recommended by the CCITT (1973) and shown in figure 2. Thirty values of the weighting coefficient at 30 frequencies are stored in each subroutine. When it is called, each subroutine calculates the weighting coefficient for the frequency of each component of the input signal (by linear interpolation with respect to the logarithm of the frequency), multiplies the coefficients by the input signal spectrum on a component-by-component basis, and calculates the total output power (by numerical integration).

4.3. Validation

All validation tests have been performed on the CDC-6600 computer of the U.S. Department of Commerce Boulder Laboratories.

As validation tests of the modulators and demodulators of the AM, FM, phase-modulation, and SSB systems, the back-to-back tests for the communication systems are simulated, in which the modulator output signal is fed directly to the demodulator input. An arbitrary modulating signal is applied to the modulator input, and after modulation and demodulation, the demodulator output signal (demodulated signal) is compared with the original modulating signal. The comparisons indicate that the corresponding spectrum components of both signals agree in at least 5 to 6 decimal digits.

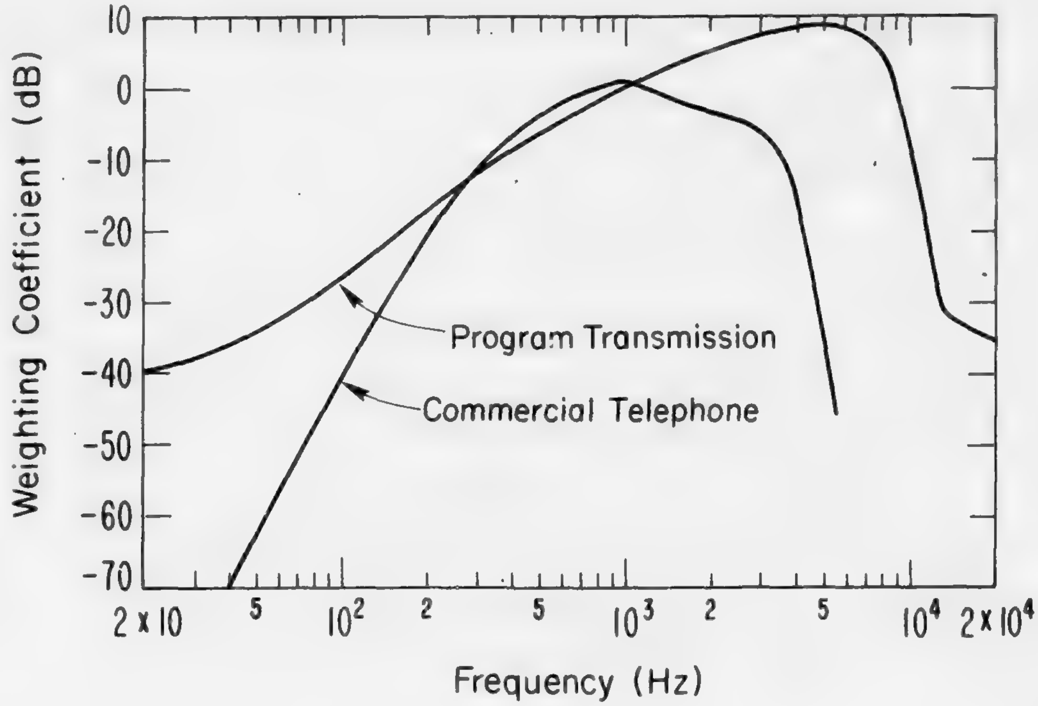


Figure 2. Frequency responses of the CCITT psophometers for commercial-telephone and program-transmission circuits.

Validation tests for the modulator and demodulator of the FSK system are performed also as the back-to-back tests. The test result indicates that the estimated bit sequence from the demodulator always agrees with the input bit sequence given to the modulator as long as the number of bits in the total time period does not exceed one half of the number of frequency spectrum components of the modulated signal.

Another validation test is performed on the FM modulator. A sinusoidal signal is given to the FM modulator input as the modulating signal, and the output spectrum is compared with the FM spectrum calculated theoretically (Schwartz, 1970). Both spectra mutually agree very well, in at least 10 decimal digits.

The Fourier-transform subprogram is also tested. The spectrum of an arbitrary signal is transformed to the time domain first, and the time-domain signal is transformed back to the frequency domain. The test result indicates that each component of the output spectrum agrees with the corresponding component of the input spectrum in 10 decimal digits or more.

5. HOW TO USE THE MODEL

5.1. General Guidelines for the User

Since the model consists of subroutine subprograms but does not include a main program, each user must write a main program that properly calls necessary subroutines. The main program is usually written in Fortran. Since computer operators will run the program on the computer, writing the main program is a major part of computer simulation for the user of the model. Although no standardized procedure exists for writing a main program, we will describe a typical procedure that can easily be followed by prospective users.

In many cases, the user must, as the first step, select necessary subroutines for the desired-signal source and the interfering-signal and/or noise sources. Selection of these subroutines depends on the problem the user wishes to solve or the system he wishes to simulate. Any subroutine that simulates a modulator or a pulse signal generator can be a desired-signal source. Any subroutine that simulates a modulator or a signal or noise generator can be an interfering-signal or a noise source. A subroutine can be used as a desired-signal source and an interfering-signal source at the same time. More than one subroutine can be used simultaneously as interfering-signal sources. The output signals from these subroutines are given as complex voltage spectra. As described earlier, the power spectrum density of the noise generator output is unity, and the output powers of all other subroutines, expressed in terms of what is used in specifying the power of an emission in the ITU Radio Regulations (1968), are equal to unity; this will make the adjustment of the signal-to-interference ratio very easy and straightforward. Although these subroutines have a number of formal parameters, it is necessary only to identify each parameter; no dimensions of arrays or numerical values of variables have to be determined at this stage.

The next step is to simulate the receiver input or to combine the interfering signal and/or noise with the desired signal. The model includes two subroutines for RF signal combiners; one linearly combines two signals with a specified ratio, and the other linearly combines three signals with two specified ratios. Again, no numerical values have to be assigned to these ratios at this stage.

The next step is to simulate radio-frequency (RF) and intermediate-frequency (IF) amplifiers with a bandpass filter. The model includes a subroutine that simulates a Chebyshev-type bandpass filter. When this subroutine is called with new values of parameters, it calls another subroutine that is also included in the model and calculates the frequency response of the bandpass filter.

The next step is to demodulate the bandpass filter output signal with a demodulator subroutine that corresponds to the desired signal. The output from the FSK demodulator is a sequence of bits, while the outputs from all other demodulators are given as complex voltage spectra in the baseband.

From the voltage spectrum in the baseband, one can derive a measure of quality of the output signal. The model includes two subroutines that simulate the psophometers for the commercial-telephone and program-transmission circuits, recommended by the CCITT (1973).

The steps described in the above five paragraphs constitute a main part of the program. This part can be repeated with various parameter values in a DO-loop or DO-loops. In some applications, this part must be executed twice, with and without the modulating signal.

Each subroutine for a modulator, except the one for the FSK modulator, accepts the input modulating signal in the form of a voltage spectrum. Also, each modulator or signal generator gives its output in the form of a voltage spectrum. The user must select suitable values for the number of frequency components, NFC, and the unit frequency spacing, UFS, in the spectrum. Selection of proper values for NFC and UFS requires some engineering judgments. The following descriptions may help the user make such selections. Each signal appearing in the model is a periodic function of time; the period of the function is represented by the reciprocal of UFS. The frequency of the highest frequency component that can appear in the model is (NFC-1)*UFS. The unit time interval of the sampling points is represented by 1/(NFC*UFS). When harmonic distortion in the demodulator output signal is discussed, selection of NFC*UFS ≥ 2*FRMX is recommended, where FRMX is the frequency at which the harmonic distortion component essentially vanishes.

In order to save computation time required for the Fourier transform, the model imposes a restriction on the value of NFC that NFC be equal to a power of two.

5.2. An Example — AM Signal With AM Interference

Let us illustrate the use of the model with a simple example. In this example, we will determine the output signal-to-interference ratio (SIR) when an amplitude-modulated (AM) desired signal is interfered with by an AM signal in the absence of noise. The output SIR depends on the modulation indexes of both the desired and interfering signals, the frequency difference between the carriers of the two signals, and the radio-frequency (RF) SIR. For simplicity we do not include the effect of a bandpass filter in this example. We measure the powers at the commercial-telephone psophometer output with and without modulation of the desired signal and calculate the output SIR as the ratio of the difference between the two output powers to the latter (i.e., without modulation).

The first step it to review briefly what subroutines are needed. Since both the desired signal and the interfering signal are AM signals, we need only the MODAM subroutine as the modulator. We can use the CBCCT2 subroutine to superpose the interfering signal upon the desired signal because we have only two signals to combine in this example. We use the DEMAM subroutine to demodulate the combined signal. To calculate the psophometrically-weighted signal power from the demodulator output signal, we use the PSPHCT subroutine that simulates the psophometer for a commercial-telephone circuit. Thus, the model includes all necessary subroutines for this example.

We will use numerals 1 and 2 for the desired signal and the interfering signal, respectively, as the endings of the variable and array names. For example, FC1 denotes the center frequency of the desired signal, and SF2 is

the name of the array for the signal that modulates the interfering signal. Then the following sequence of Fortran call statements is almost self-evident:

```
C  GENERATES THE MODULATED DESIRED SIGNAL.
      CALL MODAM (NFC,UFS,SF1,FC1,MI1,VF1)
C  GENERATES THE INTERFERING SIGNAL (MODULATED).
      CALL MODAM (NFC,UFS,SF2,FC2,MI2,VF2)
C  COMBINES THE MODULATED DESIRED SIGNAL WITH THE
C  INTERFERING SIGNAL.
      CALL CBCCT2 (NFC,VF1,VF2,SIR,VF4)
C  DEMODULATES THE COMPOSITE SIGNAL.
      CALL DEMAM (NFC,VF4,SF4)
C  APPLIES THE DEMODULATED SIGNAL TO THE PSOPHOMETER
C  INPUT.
      CALL PSPHCT (NFC,UFS,SF4,PS4)
```

(For the interpretation of parameters for these subroutines, see Appendix B.) We do not have to assign any numbers to the input parameters or determine the dimensions of arrays at this stage. In this example, we must also calculate the psophometer output power for an unmodulated desired signal. Therefore, we add to the above sequence of call statements the following statements:

```
C  GENERATES THE UNMODULATED DESIRED SIGNAL.
      CALL MODAM (NFC,UFS,SF1,FC1,Ø.Ø,VFØ)
C  COMBINES THE UNMODULATED DESIRED SIGNAL WITH THE
C  INTERFERING SIGNAL.
      CALL CBCCT2 (NFC,VFØ,VF2,SIR,VF3)
C  DEMODULATES THE COMPOSITE SIGNAL.
      CALL DEMAM (NFC,VF3,SF3)
C  APPLIES THE DEMODULATED SIGNAL TO THE PSOPHOMETER
C  INPUT.
      CALL PSPHCT (NFC,UFS,SF3,PS3)
```

(Note that generation of the interfering signal is not duplicated in this second sequence.) The order of the above nine call statements can be rearranged as long as their logical order is not violated; the only restriction is that, when the output of a subroutine A is the input of another subroutine B, the call to A must precede the call to B.

From the psophometer output powers with and without modulation, PS4 and PS3, we can calculate the output SIR, SIRŌ, as (PS4-PS3)/PS3 or as PS4/PS3-1. To avoid a possible overflow, we set SIRŌ = 10^{20} (or 200 dB) when PS3 is zero. To avoid calling the ALŌG1Ø function (common logarithmic function) with a nonpositive argument in converting SIRŌ from a power ratio to a dB value we can modify the SIRŌ value when it is less than a certain value; we set this value to be 10^{-10} (or -100 dB). These two things are minor technical points but are sometimes serious enough to frustrate the programmer. Finally, we convert SIRŌ from a power ratio to a dB value. The procedure described in this paragraph can be summarized in the following sequence of Fortran statements:

```
      IF(PS3.LE.Ø.Ø) GŌ TŌ 11
      SIRŌ = PS4/PS3-1.Ø
      IF(SIRŌ.LT.1.ØE-1Ø) SIRŌ = 1.ØE-1Ø
```

```
      GO TO 12
   11 SIRO = 1.ØE+2Ø
   12 SIRO = 1Ø.Ø*ALOG1Ø(SIRO)
```

This sequence of the Fortran statements given in the preceding two paragraphs is the core of a program that simulates reception of an AM signal interfered with by another AM signal. Except for some changes in the order of the statements, this core is common in many such programs. This sequence can be placed in a DO-loop or DO-loops when simulation is done for more than one set of parameter values.

What a user of the model does next with this core portion of the program depends on his purpose. A user may be interested in the effect of frequency difference between the desired and interfering signals, while another user may be interested in only one value of frequency difference. A user may wish to write the main program in a closed form with all input data included in the program, while another user may write the program in an open form so that the program will read in necessary data during execution.

Table 2 lists the AMAMØØ program that is written as an example for illustration purposes. This program simulates reception of an AM desired signal interfered with by another AM signal for two values of modulation index for the desired signal (MI1 = 0.5, 1.0), for two values of modulation index for the interfering signal (MI2 = 0.5, 1.0), for 31 values of frequency difference between the desired and interfering signals, and for seven values of the input RF SIR. These combinations of parameters are achieved by programming with nested 4-deep DO-loops. This program is a closed-form program and includes all necessary data in it. Values of the frequency difference (DF) and of the RF SIR are calculated in the beginning part of the program, while all other data are given to the program with the data initialization statements. This program assumes that the desired and interfering signals are modulated with 1000-Hz and 1200-Hz tones, respectively. We have selected

NFC = 128,
UFS = 200 (Hz),
FC1 = 10000 (Hz),

where NFC is the number of frequency components in the spectrum of a signal, UFS is the unit frequency spacing in the spectrum, and FC1 is the center frequency of the desired signal. Use of this UFS value limits all frequency values used in this program to integral multiples of 200 Hz. Use of this NFC value assures that all nonzero components of the modulated signals are included in the VF arrays, because the VF arrays cover the frequency range from 0 to 200 x (128-1) = 25400 Hz.

A part of the result from the AMAMØØ program is shown in table 3. The AMAMØØ program produces a total of four pages of computer printouts corresponding to the four combinations of two modulation index values for the desired signal and two modulation index values for the interfering signal. The computation time required to run the AMAMØØ program on the CDC-6600 computer was approximately 60 seconds (at a cost of approximately $8). Table 3 corresponds to the case where both indexes are equal to unity. In this table, MI1 and MI2 are the modulation indexes for the desired signal and interfering

Table 2. Fortran Listing of the AMAMØØ Program

```
      PROGRAM   AMAMCC(OUTPUT,TAPE6=OUTPUT)
C AM SYSTEM WITH AM INTERFERENCE AND NO NOISE
C THE MODULATING SIGNAL FOR THE DESIRED SIGNAL IS A 1000-HZ
C SINUSOID, AND THAT FOR THE INTERFERING SIGNAL IS A 1200-HZ
C SINUSOID.
C DECLARATION STATEMENTS
      REAL           MI1,MI2,MI1I,MI2I
      DIMENSION      MI1(2),MI2(2),DF(31),SIR(7),SIRO(7)
      COMPLEX        SF1(128),SF2(128),SF3(128),SF4(128),
     1               VF0(128),VF1(128),VF2(128),VF3(128),VF4(128)
      DATA  NM1/2/, NM2/2/, NDF/31/, NSIR/7/
      DATA  NFC/128/, UFS/200.0/, FC1/10000.0/,
     1      MI1/0.5,1.0/, MI2/0.5,1.0/,
     2      SF1/5*((0.0,0.0)),(1.0,0.0),122*((0.0,0.0))/,
     3      SF2/6*((0.0,0.0)),(1.0,0.0),121*((0.0,0.0))/
      DATA  NAME/6HAMAM00/, LUN/6/
C DATA PREPARATION
   10 DO 11  IDF=1,NDF
        DF(IDF)=200*(IDF-1)
        IF(IDF.GT.26)      DF(IDF)=1000*(IDF-21)
   11 CONTINUE
      DO 12  ISIR=1,NSIR
        SIR(ISIR)=5*(ISIR-1)
   12 CONTINUE
C SIMULATION
   20 DO 89  IM1=1,NM1
C - GENERATES THE UNMODULATED DESIRED SIGNAL.
        CALL MODAM(NFC,UFS,SF1,FC1,0.0, VF0)
C - GENERATES THE MODULATED DESIRED SIGNAL.
        MI1I=MI1(IM1)
        CALL MODAM(NFC,UFS,SF1,FC1,MI1I,VF1)
        DO 88  IM2=1,NM2
C - GENERATES THE INTERFERING SIGNAL (MODULATED).
          MI2I=MI2(IM2)
          DO 79  IDF=1,NDF
            FC2=FC1+DF(IDF)
            CALL MODAM(NFC,UFS,SF2,FC2,MI2I,VF2)
            DO 59  ISIR=1,NSIR
C - COMBINES THE UNMODULATED DESIRED SIGNAL WITH THE INTERFERING
C   SIGNAL.
              CALL CBCCT2(NFC,VF0,VF2,SIR(ISIR),VF3)
C - DEMODULATES THE COMPOSITE SIGNAL.
              CALL DEMAM(NFC,VF3,SF3)
C - APPLIES THE DEMODULATED SIGNAL TO THE PSOPHOMETER INPUT.
              CALL PSPHCT(NFC,UFS,SF3,PS3)
C - COMBINES THE MODULATED DESIRED SIGNAL WITH THE INTERFERING
C   SIGNAL.
              CALL CBCCT2(NFC,VF1,VF2,SIR(ISIR),VF4)
C - DEMODULATES THE COMPOSITE SIGNAL.
              CALL DEMAM(NFC,VF4,SF4)
C - APPLIES THE DEMODULATED SIGNAL TO THE PSOPHOMETER INPUT.
              CALL PSPHCT(NFC,UFS,SF4,PS4)
C CALCULATION
   40         IF(PS3.LE.0.0)    GO TO 42
              SIROI=PS4/PS3-1.0
              IF(SIROI.LE.1.0E-10)  SIROI=1.0E-10
              GO TO 43
```

Table 2. Continued

```
   42          SIROI=1.0E+20
   43          SIRO(ISIF)=10.0*ALOG10(SIROI)
   59        CONTINUE
C PRINTING OF THE RESULTS
   60        IF(IDF.EQ.1)  WRITE (LUN,2060)  NAME,MI1I,MI2I,SIF
             IF(MOD(IDF-1,5).LE.1)   WRITE (LUN,2061)
             WRITE (LUN,2062) DF(IDF),SIRO
   79      CONTINUE
   89    CONTINUE
   83 CONTINUE
      CALL EXIT
C FORMAT STATEMENTS
 2060 FORMAT(1H1,A6,14X,30HAM SYSTEM WITH AM INTERFERENCE//
     1   21X,4HMI1=,F5.3,5X,4HMI2=,F5.3////
     2   15X,40HOUTPUT SIGNAL-TO-INTERFERENCE RATIO (DB)/
     3   3X,2HDF,3X,16HINPUT SIR (DB) =/
     4   9X,7F7.1/)
 2061 FORMAT(1X)
 2062 FORMAT(1X,F6.0,2X,7F7.1)
      END
```

signal, respectively, and DF denotes the frequency difference in Hz between the two signals. We see from the table, for example, that, for the frequency difference (DF) of 600 Hz and an input RF signal-to-interference ratio (SIR) of 10 dB, the psophometrically weighted output SIR is 11.7 dB. Although the center frequency of the desired signal is selected as 10 kHz, the results from the AMAMØØ program including table 3 are valid for any values of the desired-signal center frequency.

5.3. Another Example — FM Distortion

As another example, we study the effects of a Chebyshev-type bandpass filter on a frequency-modulated (FM) signal. Due to the bandpass filter, the amplitude of the demodulated signal is usually suppressed, the phase is delayed, and the waveform is distorted. Assuming for simplicity that the FM signal is modulated with a single tone, we calculate the signal-suppression ratio, phase delay, and signal-to-distortion ratios (total and harmonic) for various filter parameter values and for various values of modulation index.

Simulation of this problem is very simple and straightforward. We generate the FM signal with the MODFM subroutine, apply the FM signal to the input of the bandpass filter that is simulated by the BPFCH subroutine, and demodulate the resulting signal with the DEMFM subroutine. Therefore, we can immediately write down the following sequence of call statements:

```
C  GENERATES AN FM SIGNAL.
       CALL MODFM (NFC,UFS,SF1,FC,FDV,VF1)
C  APPLIES THE FM SIGNAL TO THE BANDPASS FILTER INPUT.
       CALL BPFCH (NFC,UFS,NO,RPB,FCB,BWB,VF1,VF2)
```

Table 3. Sample Printout Produced by the AMAMØØ Program, Listed in Table 2.

AMAM00 AM SYSTEM WITH AM INTERFERENCE

MI1=1.000 MI2=1.000

OUTPUT SIGNAL-TO-INTERFERENCE RATIO (DB)

DF	INPUT SIR (DB) = 0.0	5.0	10.0	15.0	20.0	25.0	30.0
0.	2.9	8.1	13.5	19.0	24.4	29.7	34.9
200.	1.6	7.6	13.0	18.4	23.6	28.7	33.7
400.	-.2	6.0	11.7	17.3	22.6	27.7	32.8
600.	-1.2	5.8	11.7	17.2	22.4	27.6	32.6
800.	-.5	4.3	9.6	14.9	20.1	25.2	30.2
1000.	-100.0	-2.0	6.9	13.3	18.9	24.2	29.4
1200.	-100.0	1.9	9.0	14.9	20.2	25.4	30.5
1400.	-2.6	4.6	10.5	15.9	21.2	26.3	31.4
1600.	-1.2	5.2	10.9	16.4	21.7	26.8	31.8
1800.	-.9	5.1	11.0	16.5	21.8	26.9	31.9
2000.	-2.6	4.3	10.5	16.3	21.6	26.8	31.9
2200.	1.0	6.1	11.4	16.7	21.9	26.9	32.0
2400.	-1.0	5.2	11.3	16.9	22.3	27.5	32.6
2600.	-.1	6.6	12.4	17.9	23.2	28.3	33.4
2800.	.3	7.3	13.1	18.6	23.9	29.0	34.1
3000.	.2	7.6	13.6	19.2	24.5	29.6	34.7
3200.	.1	8.2	14.5	20.1	25.5	30.6	35.7
3400.	.4	9.2	15.6	21.2	26.4	31.5	36.6
3600.	.2	10.0	16.8	22.4	27.6	32.7	37.8
3800.	1.5	10.8	17.8	23.6	29.0	34.2	39.2
4000.	1.6	11.4	18.8	24.8	30.2	35.4	40.5
4200.	1.7	11.9	19.8	26.0	31.5	36.7	41.8
4400.	1.4	12.5	21.0	27.5	33.1	38.3	43.4
4600.	1.8	12.9	21.9	28.6	34.3	39.6	44.7
4800.	1.8	13.1	22.8	30.0	36.0	41.3	46.5
5000.	1.8	13.6	23.8	31.8	38.2	43.8	49.0
6000.	1.7	13.9	25.7	36.3	46.4	56.0	64.7
7000.	2.0	14.5	26.1	36.5	46.7	56.8	66.9
8000.	1.8	13.8	25.5	36.3	46.7	56.8	66.9
9000.	1.9	13.8	25.5	36.3	46.6	56.8	66.9
10000.	1.9	13.6	25.5	36.3	46.6	56.8	66.8

```
C  DEMODULATES THE FILTER OUTPUT SIGNAL.
      CALL DEMFM (NFC,UFS,VF2,FC,SF2)
```

(For the interpretation of parameters for these subroutines, see Appendix B.) Again, no numbers have to be assigned to any parameters at this stage.

The above sequence of call statements leaves the spectrum of the demodulated signal in the SF2 array as the result of simulation. Although it may look a little complicated, calculation of the final result from the result of the simulation left in SF2 is by no means difficult once one understands the physical phenomena.

If the modulated signal passes through no bandpass filter, the SF2 array will coincide with the SF1 array that represents the modulating signal spectrum. If the modulated signal passes through a bandpass filter, on the other hand, SF2 generally differs from SF1. Any difference between SF2 and SF1 is the result of the effect of the bandpass filter. The way of representing the difference between the two spectra may not be unique but, when modulation is with a single tone as assumed in this example, the following representation seems reasonable.

Since we have assumed modulation with a single tone, the SF1 array has, as the only nonzero element, its second element that corresponds to the fundamental frequency component. The second element of the demodulated signal spectrum SF2 is generally different from that of SF1 both in amplitude and phase, i.e., it is usually suppressed in amplitude and delayed in phase. We define the signal-suppression ratio and phase delay to represent these differences. In addition to the differences in the amplitude and phase of the fundamental-frequency component, we can define the total distortion, as the ratio of the power of the fundamental component to the total power of all harmonic components, and also the harmonic distortions, each as the ratio of the power of the fundamental-frequency component to the power of each harmonic component.

Harmonic distortion powers and/or the total distortion power can be zero. To avoid calling the ALŌG1Ø function (common logarithmic function) with a zero argument in converting the signal-to-distortion ratio from a power ratio to a dB value, we set the ratio to be 1000 dB when the distortion power is zero.

The FMDFØØ program that is programmed for illustration purposes is listed in table 4. It is again a closed-form program; all data are either given with the data initialization statements or calculated during execution. We have selected

NFC = 128,
UFS = 1.0,
FC = 64.0,
FM = 1.0,

where NFC is the number of frequency components in the spectrum of a signal, UFS is the unit frequency spacing in the spectrum, FC is the center frequency of the modulated signal, and FM is the frequency of the modulating tone. For simplicity we have normalized all frequencies and bandwidths in the program

Table 4. Fortran Listing of the FMDFØØ Program

```
      PROGRAM   FMDF00(OUTPUT,TAPE6=OUTPUT)
C FM DISTORTION CAUSED BY A BANDPASS FILTER OF CHEBYSHEV TYPE
C DECLARATION STATEMENTS
      COMPLEX      SF1(128),VF1(128),VF2(128),SF2(128)
      DIMENSION    PF(128),DS(10)
      DIMENSION    MI(4),NOB(2),RPB(3),CFBW(7),CFDF(4)
      REAL         MI
      DATA  SF1/(0.0,0.0),(1.0,0.0),126*((0.0,0.0))/
      DATA  NFC/128/, UFS/1.0/, FC/64.0/, FM/1.0/,
     1      MI/1.0,2.0,5.0,10.0/,
     2      NOB /2,4/,
     3      RPB /0.0,1.0,2.0/,
     4      CFBW/0.8,1.0,1.2,1.4,1.6,1.8,2.0/,
     5      CFDF/0.0,0.1,0.2,0.3/
      DATA  NAME/6HFMDF00/, LUN/6/
C SIMULATION
   10 DO 89  IMI=1,4
C - GENERATES AN FM SIGNAL.
        FDV=FM*MI(IMI)
        P0=FDV*FDV/2.0
        CALL MODFM(NFC,UFS,SF1,FC,FDV,VF1)
C - DETERMINES PARAMETERS FOR A BANDPASS FILTER.
        BWCS=2.0*(FM+FDV)
        DO 88  INOB=1,2
          NOB0=NOB(INOB)
          DO 87  IRPB=1,3
            RPB0=RPB(IRPB)
            IF(NOB0.EQ.1.AND.RPB0.NE.0.0)      GO TO 87
            DO 69  IBWB=1,7
              BWB=BWCS*CFBW(IBWB)
              DO 59  IFCB=1,4
                OFC=BWB*CFDF(IFCB)
                FCB=FC+OFC
C - APPLIES THE FM SIGNAL TO THE BANDPASS FILTER INPUT.
                CALL BPFCH(NFC,UFS,NOB0,RPB0,FCB,BWB,VF1,VF2)
C - DEMODULATES THE FILTER OUTPUT SIGNAL.
                CALL DEMFM(NFC,UFS,VF2,FC,SF2)
C CALCULATION
C - CALCULATES THE FUNDAMENTAL FREQUENCY COMPONENT OF THE OUTPUT
C   SIGNAL POWER.
   30           P1=(REAL(SF2(2))**2+AIMAG(SF2(2))**2)/2.0
C - CALCULATES THE HARMONIC COMPONENTS OF THE OUTPUT SIGNAL
C   POWER AND THE TOTAL DISTORTION POWER.
                PT=0.0
                DO 31  I=3,NFC
                  PF(I)=(REAL(SF2(I))**2+AIMAG(SF2(I))**2)/2.0
                  PT=PT+PF(I)
   31           CONTINUE
C - CALCULATES THE SIGNAL SUPPRESSION RATIO.
                SSR=-10.0*ALOG10(P1/P0)
C - CALCULATES THE PHASE DELAY.
                PHS=ATAN2(AIMAG(SF2(2)),REAL(SF2(2)))
C - CALCULATES THE SIGNAL-TO-DISTORTION RATIO.
                IF(PT.GT.0.0)        DT=-10.0*ALOG10(PT/P1)
                IF(PT.LE.0.0)        DT=1000.0
                DO 32  I=3,6
                  IF(PF(I).GT.0.0)  DS(I-1)=
```

Table 4. Continued

```
      1                                -10.0*ALCG10(PF(I)/F1)
                  IF(PF(I).LE.0.0)  DS(I-1)=10C0.0
     32         CONTINUE
C PRINTING OF THE RESULTS
     40         IF(IBWB.EQ.1.AND.IFCB.EQ.1)
      1            WRITE (LUN,2040) NAME,MI(IMI),NC30,RPB0
                IF(IFCB.GT.1)  GC TO 42
     41         WRITE (LUN,2041) BWB,DFC,SSR,PHS,DT,
      1                          (DS(IH),IH=2,5)
                GO TC 59
     42         WRITE (LUN,2042) DFC,SSR,PHS,DT,(DS(IH),IH=2,5)
     59       CONTINUE
     69     CONTINUE
     87   CONTINUE
     88  CONTINUE
     89 CONTINUE
        CALL EXIT
C FORMAT STATEMENTS
   2040 FORMAT(1H1,A6,8X,36HFM DISTORTION DUE TO BANDFASS FILTEF//
      1   15X,4HMI =,F5.1,6X,5HNOB =,I2,6X,9HRPB(DB) =,F4.1////
      2   21X,3HSSR,2X,5HPHASE,6X,2HDT,5X,2HD2,5X,2HD3,
      3   5X,2HD4,5X,2HD5/
      4   3X,3HBWB,5X,3HDFC/
      5   20X,4H(DB),2X,5H(RAD),2X,5(3X,4H(DB)))
   2041 FORMAT(1X/1X,F6.3,F8.3,2X,2F7.2,2X,5F7.1)
   2042 FORMAT(1X,6X, ,F8.3,2X,2F7.2,2X,5F7.1)
        END
```

to FM by taking FM = 1.0; this convenient normalization is possible when a program does not simulate the psophometer that is dependent on the absolute frequency values. This program performs simulation for a total of 24 combinations of four values of modulation index (MI = 1.0, 2.0, 5.0, 10.0), two values of order of the bandpass filter (NŌB = 2, 4), and three values of ripple in the filter passband (RPF = 0.0, 1.0, 2.0 dB). For each combination, FMDFØØ calculates seven values of filter bandwidth, BWB, i.e., 90%, 100%, 120%, 140%, 160%, 180%, and 200% of the Carson bandwidth, 2*(MI + 1.0)*FM. For each BWB value, FMDFØØ further calculates four values of detuning frequency, DFC, as 0%, 10%, 20%, and 30% of the BWB value. (Detuning frequency is the difference in the center frequencies of the bandpass filter and the signal.) For each combination of MI, NŌB, and RPB and for each value of BWB and DFC, the FMDFØØ program performs simulation and calculates the signal-suppression ratio (SSR), phase (PHASE), total distortion (DT), and the second through the fifth harmonic distortions (D2, D3, D4, D5).

A part of the result from the FMDFØØ program is shown in table 5. This table represents the result for the modulation index of ten (MI = 10.0), the fourth-order bandpass filter (NŌB = 4), and no ripple in the filter response (RPB = 0.0 dB). The FMDFØØ program yields 23 more pages of similar printouts for different combinations. The computation time required for the FMDFØØ program on the CDC-6600 computer was approximately 48 seconds.

Table 5 shows the signal-suppression ratio (SSR), the phase (PHASE), the total distortion (DT), and the four harmonic distortions (D2, D3, D4, and D5),

Table 5. Sample Printout Produced by the FMDFØØ Program, Listed in Table 4.

FMDF00 FM DISTORTION DUE TO BANDPASS FILTER

MI = 10.0 NOB = 4 RPB(DB) = 0.0

BWB	DFC	SSR (DB)	PHASE (RAD)	CT (DB)	D2 (DB)	D3 (DB)	D4 (DB)	D5 (DB)
17.600	0.000	.08	-.35	26.1	268.8	27.6	270.6	35.1
	1.760	.06	-.35	25.6	36.6	31.1	30.7	33.9
	3.520	.02	-.34	21.1	36.8	30.9	24.6	31.0
	5.280	-.02	-.32	15.3	32.6	24.0	24.3	27.3
22.000	0.000	.03	-.27	29.6	272.2	30.1	271.5	42.0
	2.200	.04	-.27	29.8	32.8	32.3	41.4	45.1
	4.400	.03	-.27	27.4	31.1	40.6	32.4	35.9
	6.600	.01	-.27	23.7	37.5	28.5	28.4	37.5
26.400	0.000	.01	-.21	34.2	271.7	34.4	270.5	47.9
	2.640	.02	-.22	31.3	34.1	35.2	47.3	51.6
	5.280	.03	-.23	29.4	30.6	42.7	40.0	40.9
	7.920	.02	-.24	28.0	34.3	33.6	32.2	42.8
30.800	0.000	.00	-.18	38.8	272.6	38.9	285.8	54.3
	3.080	.01	-.18	34.2	36.6	38.4	50.6	56.4
	6.160	.02	-.20	30.9	31.6	41.4	46.8	46.2
	9.240	.02	-.21	30.4	33.2	39.4	35.9	44.9
35.200	0.000	.00	-.15	43.0	270.7	43.1	270.8	60.6
	3.520	.01	-.16	37.1	39.2	41.6	54.1	61.0
	7.040	.02	-.17	32.5	33.2	42.0	52.3	51.4
	10.560	.02	-.18	32.0	33.2	45.9	39.5	47.2
39.600	0.000	.00	-.14	46.7	273.1	46.7	273.2	66.5
	3.960	.00	-.14	39.8	41.7	44.8	57.7	65.5
	7.920	.01	-.15	34.2	34.9	43.4	56.2	56.3
	11.880	.02	-.16	33.1	33.7	53.9	43.0	49.9
44.000	0.000	.00	-.12	49.9	271.2	49.9	277.1	71.8
	4.400	.00	-.12	42.3	43.9	47.7	61.3	69.8
	8.800	.01	-.13	36.0	36.6	45.2	59.2	61.1
	13.200	.02	-.15	34.2	34.5	64.5	46.3	52.8

for four values of detuning frequency (DFC) and seven values of the filter bandwidth (BWB). Since the frequency of the modulating tone, FM, is taken to be unity in this program, we can consider that all frequencies and bandwidths are normalized to the modulating-tone frequency. Although the center frequency of the modulated signal is selected as 64.0 times the modulating-tone frequency, the results from the FMDFØØ program including table 5 are valid for any values of the modulated-signal center frequency.

It is expected that no even-order distortion will be produced when the bandpass filter is tuned to the modulated signal. Table 5 indicates that D2

and D4 are greater than 260 dB for DFC = 0.0. This result can be interpreted to mean that our program simulates a laboratory test in which the background noise level in measurements of harmonic distortions is at least 260 dB below the desired output signal level (or the fundamental-frequency component of the output signal).

5.4. Third Example — FM Noise

As the third and last example, we simulate reception of an FM signal in the presence of white Gaussian noise. This is a classic problem. As in the first example described in section 5.2, we simulate reception of two desired signals with and without modulation, determine two corresponding psophometrically-weighted powers, and calculate the output signal-to-noise ratio (SNR) as the ratio of the difference between the two output powers to the output power without modulation. Programming the core of the program for this example is very similar to programming the first example described in 5.2. The main difference between this example and the first example, other than the differences in modulator, signal generator, and demodulator, is the inclusion of a bandpass filter in this example. Since the interfering signal is noise in this example, the result largely depends on the RF (or IF) bandwidth; in other words, inclusion of a bandpass filter is essential in this simulation.

The FMWGØØ program that is written for illustration purposes is listed in table 6. In this program, we have selected

NFC = 128,
UFS = 500 (Hz),
FC = 30000 (Hz),

where NFC is the number of frequency components in the spectrum of a signal, UFS is the unit frequency spacing in the spectrum, and FC is the center frequency of the modulated signal. We have also assumed a baseband bandwidth of 3 kHz, modulation with a single tone of frequency 1 kHz, and a fourth-order bandpass filter with the Carson bandwidth and with no ripple in the passband. The FMWGØØ program performs simulation for nine values of peak frequency deviation and for 11 values of the input signal-to-noise-density ratio (SNDR), which is defined as the ratio of signal power (in watts) to noise power density (in watts/hertz or joules). Since the unit frequency spacing (UFS) is 500 Hz in this program, this program simulates measurements of output SNR in an interval of 2 ms (or 2 cycles of the modulating signal). This short interval of measurement will result in fluctuations in the output SNR values and therefore irregularities in the curves for the output SNR values vs. the input SNR (or SNDR) values.

The result of the FMWGØØ program is shown in table 7. The computation time required to run the FMWGØØ program on the CDC-6600 computer was approximately 15 seconds. In table 7, FDV denotes the peak frequency deviation in Hz. The output SNR values given in table 7 exhibit some irregularities, especially in the left portion of the table; the output SNR should be a monotonic increasing function of the input SNDR for a fixed FDV value theoretically, while the output SNR shown in table 7 is not monotonic. This

Table 6. Fortran Listing of the FMWGØØ Program

```
      PROGRAM  FMWG00(OUTPUT,TAPE6=OUTPUT)
C FM SYSTEM WITH WHITE GAUSSIAN NOISE
C THE INPUT SIGNAL-TO-NOISE-DENSITY RATIO IS THE RATIO OF SIGNAL
C POWER (WATTS) TO NOISE POWER DENSITY (WATTS/HERTZ).
C THE OUTPUT SIGNAL-TO-NOISE RATIO IS CALCULATED AS THE RATIO OF
C OUTPUT POWER WITH MODULATION TO OUTPUT POWER WITHOUT MODULA-
C TION MINUS ONE.
C DECLARATION STATEMENTS
      DIMENSION   FDV(9),SNDR(11),SNRO(11)
      COMPLEX     SF1(128),SF4(128),SF5(128),
     1            VF1(128),VF2(128),VF3(128),VF4(128),VF5(128)
      DATA  NFD/9/, NSNR/11/
      DATA  NFC/128/, UFS/500.0/, FC/30000.0/,
     1      FDV/10000.,5000.,3000.,2000.,1000.,500.,300.,200.,
     2          100./,
     3      SF1/2*((0.0,0.0)),(1.0,0.0),125*((0.0,0.0))/
      DATA  NOB/4/, RPB/0.0/
      DATA  SNROMN/1.02E-10/
      DATA  NAME/6HFMWG00/, LUN/6/
C DATA PREPARATION
   10 DO 11  ISNR=1,NSNR
        SNDR(ISNR)=2*(ISNR-1)+40
   11 CONTINUE
C SIMULATION
C - GENERATES UNMODULATED SIGNAL.
   20 CALL MODFM(NFC,UFS,SF1,FC,0.0,VF2)
C - GENERATES WHITE GAUSSIAN NOISE.
      CALL SGWGN(NFC,UFS,2,NFC,VF3)
      DO 89  IFD=1,NFD
C - GENERATES AN FM SIGNAL.
        FDI=FDV(IFD)
        CALL MODFM(NFC,UFS,SF1,FC,FDI,VF1)
C - ADJUSTS THE FILTER BANDWIDTH TO THE CARSON BANDWIDTH.
        BWB=2.0*(3000.0+FDI)
        DO 59  ISNR=1,NSNR
C - COMBINES THE FM SIGNAL WITH THE NOISE.
          CALL CBCCT2(NFC,VF1,VF3,SNDR(ISNR),VF4)
C - APPLIES THE COMPOSITE SIGNAL TO THE BANDPASS FILTER INPUT.
          CALL BPFCH(NFC,UFS,NOB,RPB,FC,BWB,VF4,VF4)
C - DEMODULATES THE FILTER OUTPUT SIGNAL.
          CALL DEMFM(NFC,UFS,VF4,FC,SF4)
C - APPLIES THE DEMODULATED SIGNAL TO THE PSOPHOMETER INPUT.
          CALL PSPHCT(NFC,UFS,SF4,PS4)
C - COMBINES THE UNMODULATED SIGNAL WITH THE NOISE.
          CALL CBCCT2(NFC,VF2,VF3,SNDR(ISNR),VF5)
C - APPLIES THE COMPOSITE SIGNAL TO THE BANDPASS FILTER INPUT.
          CALL BPFCH(NFC,UFS,NOB,RPB,FC,BWB,VF5,VF5)
C - DEMODULATES THE FILTER OUTPUT SIGNAL.
          CALL DEMFM(NFC,UFS,VF5,FC,SF5)
C - APPLIES THE DEMODULATED SIGNAL TO THE PSOPHOMETER INPUT.
          CALL PSPHCT(NFC,UFS,SF5,PS5)
C CALCULATION
          IF(PS5.LE.0.0)  GO TO 42
          SNROI=PS4/PS5-1.0
          IF(SNROI.LE.SNROMN)  SNROI=SNROMN
          GO TO 43
   42     SNROI=1.0E+20
```

Table 6. Continued

```
   43     SNRO(ISNR)=10.0*ALOG10(SNROI)
   59   CONTINUE
C PRINTING OF THE RESULTS
   60   IF(IFD.EQ.1)      WRITE (LUN,2060)  NAME,SNOR
        WRITE (LUN,2061) FDI,SNRO
   89 CONTINUE
      CALL EXIT
C FORMAT STATEMENTS
 2060 FORMAT(1H1,A6,8X,35HFM SYSTEM WITH WHITE GALSSIAN NCISE//
     1   15X,28HBASEBAND BANDWIDTH = 3000 HZ/
     2   15X,38HMODULATING SIGNAL  ---  A 1000-HZ TONE////
     3   27X,33HOUTPUT SIGNAL-TO-NOISE RATIO (DB)/
     4   3X,3HFDV,3X,17HINPUT SNOR (DB) =/
     5   9X,11F6.1/)
 2061 FORMAT(1X/1X,F6.0,2X,11F6.1)
      END
```

Table 7. Computer Printout Produced by the FMWGØØ Program, Listed in Table 6

FMWG00 FM SYSTEM WITH WHITE GAUSSIAN NOISE

BASEBAND BANDWIDTH = 3000 HZ
MODULATING SIGNAL --- A 1000-HZ TONE

OUTPUT SIGNAL-TO-NOISE RATIO (DB)

FDV	INPUT SNOR (DB) = 40.0	42.0	44.0	46.0	48.0	50.0	52.0	54.0	56.0	58.0	60.0
10000.	7.4	8.0	5.6	19.3	24.3	29.9	32.6	34.8	37.0	39.0	41.1
5000.	9.6	16.0	17.9	20.3	22.7	24.9	27.0	29.1	31.1	33.2	35.2
3000.	7.0	10.7	13.4	15.8	18.1	20.2	22.4	24.5	26.5	28.6	30.6
2000.	5.7	7.8	10.6	12.7	14.9	16.9	19.0	21.1	23.1	25.2	27.2
1000.	-1.6	-.1	2.9	5.5	8.0	10.3	12.6	14.7	16.9	19.0	21.1
500.	5.7	-1.1	.1	2.0	4.0	6.0	8.0	9.9	11.9	13.8	15.8
300.	-4.7	-17.3	-99.9	-8.6	-3.5	-.2	2.5	4.8	7.0	9.1	11.2
200.	-13.6	-14.0	-13.1	-10.5	-7.5	-4.7	-2.1	.4	2.7	4.9	7.1
100.	17.8	-99.9	-99.9	-99.9	-99.9	-24.1	-9.3	-5.4	-2.6	-.3	1.9

irregularity is considered to stem from the choice of a large UFS value, as discussed in the preceding paragraph. It is expected that this irregularity can be reduced by taking a small UFS value (and thus increasing the measurement interval) but with a corresponding increase in the computation time. Although the center frequency of the modulated signal is selected as 30 kHz in this program, the result shown in table 7 is independent of the modulated-signal center frequency.

6. CONCLUDING REMARKS

We have started with elementary concepts concerning computer simulation of communication systems and a simulation model, derived guidelines for developing a simulation model, described a model that has been developed along these guidelines, and explained how to use the model. The model described in this report is still under development. For further development of the model, we make the following remarks:

The model presented here includes modulators and demodulators of the AM, FM, FSK, phase-modulation (analog), and SSB systems. Because of its increased use, inclusion of a phase-shift-keying (PSK) system is highly desirable. It is anticipated that inclusion of PSK will necessitate simulation of phase-lock loop (PLL), which has a wide field of application.

At the present time, no baseband signal generator is included in the model. The addition of generators of some typical baseband signals, such as a voice signal, will be desirable.

Only a Gaussian noise generator is included in the model as a noise generator. Inclusion of non-Gaussian noise generators for atmospheric noise and man-made noise is expected to make the model more versatile. It is also considered desirable to include interference generators that simulate unintentional emissions from industry or various types of home equipment.

Only a Chebyshev-type bandpass filter is included in the model. It is expected that inclusion of an elliptic-function-type filter will make the model more versatile and better able to simulate the transfer characteristics of particular receiving systems.

In addition, techniques for specifying more specific and detailed performance measures (such as articulation scores for voice signals, bit error rates for digital signals, direction finding errors, etc.) need to be developed.

Inclusions of new types of modulators and demodulators, signal and noise generators, bandpass and lowpass filters, etc., described above can be made by adding to the model additional subroutines that simulates these system components. Those subroutines already programmed and presented in this report need no modifications.

Articulation index (AI) is sometimes used as a measure of voice intelligibility (ANSI, 1970). In our model, however, we have intentionally omitted AI because the validity of AI for radio-interference problems is questionable.

7. REFERENCES

Abramowitz, M., and I. A. Stegun (1964), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, U.S. National Bureau of Standards Applied Mathematics Series (AMS) 55 (U.S. Government Printing Office, Washington, D.C. 20402), ch. 22.

Akima, Hiroshi (1963), Theoretical studies on signal-to-noise characteristics of an FM system, IEEE Trans. Space Electronics and Telemetry, SET-9, 101-108.

ANSI (American National Standards Institute) (1966), ANSI Standard Fortran, Publication X3.9-1966, ANSI, New York, N.Y. Also reproduced in "History and summary of FORTRAN standardization development for the ASA," by W. P. Heising, Commun. ACM, 7, 590-625, October 1964.

ANSI (1970), Methods for calculation of the articulation index, Publication S3.5-1969, ANSI, New York, N.Y.

Box, G., and M. Muller (1958), A note on the generation of normal deviates, Ann. Math. Stat., 28, 610-613.

CCIR (International Radio Consultative Committee) (1975), Non-coherent receiver performance model, Report 520, CCIR Green Book, Vol. I, pp. 192-205, International Telecommunication Union (ITU), Geneva, Switzerland.

CCITT (International Consultative Committee on Telegraph and Telephone)(1973), Psophometers (Apparatus for the Objective Measurement of Circuit Noise), Recommendation P.53, CCITT Green Book, Vol. V, 87-95, ITU, Geneva Switzerland.

Cochran, W. T., J. W. Cooley, D. L. Favin, H. D. Helms, R. A. Kaenel, W. W. Lang, G. C. Maling, Jr., D. E. Nelson, C. M. Rader, and P. D. Welch (1967), "What is the fast Fourier transform?" Proc. IEEE, 55, 1664-1674.

Cooley, J. W., and J. W. Tukey (1965), An algorithm for the machine calculation of complex Fourier series, Math. of Comput., 19, 297-301.

ITU (International Telecommunication Union) (1968), Radio Regulations (Appendix 1 in particular), ITU, Geneva, Switzerland.

Schwartz, M. (1970), Information Transmission, Modulation, and Noise (McGraw-Hill, New York, N.Y.), ch. 4.

Storer, J. E. (1957), Passive Network Synthesis (McGraw-Hill, New York, N.Y.), ch. 30.

Stumpers, F. L. H. M. (1948), Theory of frequency-modulation noise, Proc. IRE, 36, 1081-1092.

Wozencraft, J. M., and I. M. Jacobs (1965), Principles of Communication Engineering (John Wiley & Sons, New York, N.Y.), ch. 8.

APPENDIX A

FORTRAN LISTINGS OF THE MODEL

Complete Fortran listings of all subprograms included in the model are given in this appendix in alphabetical order. All subroutines are written in ANSI Standard Fortran.

The RANF function called by some subroutines is a random-number generator. Repeated calls to RANF generates a sequence of random numbers uniformly distributed in a range extending from 0 to 1. If the computer system library does not include this function, the user of the model must supply it.

In addition, the following intrinsic and basic library functions are called by some subroutines. They are included in most computer systems that accept ANSI Standard Fortran.

Name	Description
ABS	Absolute value of a real number.
AINT	Integer part of a real number (truncation).
AMOD	Real modulo function.
CMPLX	A complex number from two real numbers.
FLOAT	Conversion from integer to real.
ALOG	Natural logarithmic function.
ATAN2	Arctangent function.
COS	Cosine function.
EXP	Exponential function.
SIN	Sine function.
SQRT	Square-root function.

```
      SUBROUTINE  BPFCH (NFC,UFS,NOB,RPB,FCB,BWB,VF1,VF2)
C THIS SUBROUTINE SIMULATES A BANDPASS FILTER (BPF) OF CHEBYSHEV
C TYPE.
C THE INPUT PARAMETERS ARE
C      NFC = NUMBER OF FREQUENCY COMPONENTS
C            (MUST BE POSITIVE BUT NOT GREATER THAN 2048),
C      UFS = UNIT FREQUENCY SPACING (MUST BE POSITIVE),
C      NOB = ORDER OF THE BPF
C            (MUST NOT BE NEGATIVE OR GREATER THAN 100),
C      RPB = RIPPLE (DB) WITHIN THE PASSBAND OF THE BPF
C            (MUST BE NONNEGATIVE),
C      FCB = CENTER FREQUENCY OF THE BPF
C            (MUST BE POSITIVE),
C      BWB = 3-DB BANDWIDTH OF THE BPF (MUST BE POSITIVE),
C      VF1 = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C            SPECTRUM COMPONENTS OF THE INPUT VOLTAGE.
C THE OUTPUT PARAMETER IS
C      VF2 = COMPLEX ARRAY OF DIMENSION NFC WHERE THE
C            SPECTRUM COMPONENTS OF THE OUTPUT VOLTAGE
C            ARE TO BE STORED.
C THIS SUBROUTINE CALLS THE FRBPFC SUBROUTINE.
C DECLARATION STATEMENTS
      COMPLEX      VF1(NFC),VF2(NFC)
      COMPLEX      HFB(2048)
      DATA  NFCMX/2048/, NOBMX/100/
      DATA  NFCPV/0/, UFSPV/0.0/,NOBPV/-1/, RPBPV/-1.0/,
     1      FCBPV/0.0/, BWBPV/0.0/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 NFC0=NFC
      UFS0=UFS
      NOB0=NOB
      RPB0=RPB
      FCB0=FCB
      BWB0=BWB
      IF(NFC0.LE.0.OR.NFC0.GT.NFCMX)      GO TO 90
      IF(UFS0.LE.0.0)      GO TO 90
      IF(NOB0.LT.0.OR.NOB0.GT.NOBMX)      GO TO 90
      IF(RPB0.LT.0.0)      GO TO 90
      IF(FCB0.LT.0.0)      GO TO 90
      IF(BWB0.LT.0.0)      GO TO 90
      IF(NOB0.EQ.0)        GO TO 40
C CALCULATES FREQUENCY RESPONSE OF THE BPF WHEN NECESSARY.
   20 IF(NFC0.EQ.NFCPV.AND.UFS0.EQ.UFSPV.AND.
     1   NOB0.EQ.NOBPV.AND.RPB0.EQ.RPBPV.AND.
     2   FCB0.EQ.FCBPV.AND.BWB0.EQ.BWBPV)      GO TO 30
      CALL FRBPFC(NOB0,RPB0,FCB0,BWB0,UFS0,1,NFC0,HFB)
C CALCULATES OUTPUT SPECTRUM.
C - NOB.NE.0
   30 DO 31  I=1,NFC0
         VF2(I)=HFB(I)*VF1(I)
   31 CONTINUE
      RETURN
C - NOB.EQ.0
   40 DO 41  I=1,NFC0
         VF2(I)=VF1(I)
   41 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) NFC0,UFS0,NOB0,RPB0,FCB0,BWB0
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***   IMPROPER INPUT VALUE(S)./
     1   3X,5HNFC =,I6,5X,5HUFS =,E11.3,5X,5HNOB =,I6,
```

```
      2   5X,5HRPB =,E11.3,5X,5HFCB =,E11.3,5X,5HBWB =,E11.3/
      3   35H ERROR DETECTED IN ROUTINE   BPFCH /)
          END


          SUBROUTINE  CBCCT2(NFC,VF1,VF2,RDB,VF3)
    C THIS SUBROUTINE SIMULATES A RADIO-FREQUENCY (RF) COMBINING
    C CIRCUIT THAT LINEARLY COMBINES TWO RF SIGNALS WITH A SPECI-
    C FIED RATIO.
    C THE LEVEL OF THE FIRST INPUT SIGNAL IS UNCHANGED IN THIS
    C CIRCUIT.
    C THE INPUT PARAMETERS ARE
    C     NFC = NUMBER OF FREQUENCY COMPONENTS IN THE SPECTRUM
    C           OF EACH INPUT SIGNAL,
    C     VF1 = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
    C           VOLTAGE SPECTRUM COMPONENTS OF THE FIRST INPUT
    C           SIGNAL,
    C     VF2 = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
    C           VOLTAGE SPECTRUM COMPONENTS OF THE SECOND INPUT
    C           SIGNAL,
    C     RDB = SIGNAL-TO-INTERFERENCE RATIO OR SIGNAL-TO-NOISE-
    C           DENSITY RATIO BETWEEN THE FIRST AND THE SECOND
    C           INPUT SIGNALS IN DB.
    C THE OUTPUT PARAMETER IS
    C     VF3 = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
    C           SPECTRUM COMPONENTS OF THE COMPOSITE SIGNAL ARE
    C           TO BE STORED.
    C DECLARATION STATEMENTS
          DIMENSION   VF1(64),VF2(64),VF3(64)
          DATA  RPV/0.0/, VR/1.0/
    C PRELIMINARY PROCESSING
       10 N0=NFC
          R0=RDB
          NT2=N0*2
    C CONVERTS SIGNAL-TO-INTERFERENCE RATIO OR SIGNAL-TO-NOISE-
    C DENSITY RATIO FROM DB TO VOLTAGE RATIO WHEN NECESSARY.
       20 IF(R0.EQ.RPV)  GO TO 30
          RPV=R0
          VR=EXP(-0.1151292546*R0)
    C LINEARLY COMBINES THE SPECTRA OF THE TWO SIGNALS ON
    C COMPONENT-BY-COMPONENT BASIS.
       30 DO 31  J=1,NT2
            VF3(J)=VF1(J)+VR*VF2(J)
       31 CONTINUE
          RETURN
          END


          SUBROUTINE  CBCCT3(NFC,VF1,VF2,VF3,R12DB,R13DB,VF4)
    C THIS SUBROUTINE SIMULATES A RADIO-FREQUENCY (RF) COMBINING
    C CIRCUIT THAT LINEARLY COMBINES THREE RF SIGNALS WITH SPECI-
    C FIED RATIOS.
    C THE LEVEL OF THE FIRST INPUT SIGNAL IS UNCHANGED IN THIS
    C CIRCUIT.
    C THE INPUT PARAMETERS ARE
    C     NFC   = NUMBER OF FREQUENCY COMPONENTS IN THE SPECTRUM
    C             OF EACH INPUT SIGNAL,
    C     VF1   = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
    C             VOLTAGE SPECTRUM COMPONENTS OF THE FIRST INPUT
    C             SIGNAL,
    C     VF2   = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
    C             VOLTAGE SPECTRUM COMPONENTS OF THE SECOND INPUT
```

```
C                 SIGNAL,
C       VF3   = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C                 VOLTAGE SPECTRUM COMPONENTS OF THE THIRD INPUT
C                 SIGNAL,
C       R12DB = SIGNAL-TO-INTERFERENCE RATIO OR SIGNAL-TO-
C                 NOISE-DENSITY RATIO BETWEEN THE FIRST AND THE
C                 SECOND INPUT SIGNALS IN DB,
C       R13DB = SIGNAL-TO-INTERFERENCE RATIO OR SIGNAL-TO-
C                 NOISE-DENSITY RATIO BETWEEN THE FIRST AND THE
C                 THIRD INPUT SIGNALS IN DB.
C THE OUTPUT PARAMETER IS
C       VF4   = COMPLEX ARRAY OF DIMENSION NFC WHERE THE
C                 VOLTAGE SPECTRUM COMPONENTS OF THE COMPOSITE
C                 SIGNAL ARE TO BE STORED.
C DECLARATION STATEMENTS
      DIMENSION   VF1(64),VF2(64),VF3(64),VF4(64)
      DATA  R2PV/0.0/, R3PV/0.0/, VR2/1.0/, VR3/1.0/
C PRELIMINARY PROCESSING
   10 NO=NFC
      R2=R12DB
      R3=R13DB
      NT2=NO*2
C CONVERTS SIGNAL-TO-INTERFERENCE RATIOS AND/OR SIGNAL-TO-NOISE-
C DENSITY RATIOS FROM DB TO VOLTAGE RATIOS WHEN NECESSARY.
   20 IF(R2.EQ.R2PV)      GO TO 21
      R2PV=R2
      VR2=EXP(-0.1151292546*R2)
   21 IF(R3.EQ.R3PV)      GO TO 30
      R3PV=R3
      VR3=EXP(-0.1151292546*R3)
C LINEARLY COMBINES THE SPECTRA OF THE THREE SIGNALS ON
C COMPONENT-BY-COMPONENT BASIS.
   30 DO 31  J=1,NT2
        VF4(J)=VF1(J)+VR2*VF2(J)+VR3*VF3(J)
   31 CONTINUE
      RETURN
      END



      SUBROUTINE  DEMAM (NFC,VF,SF)
C THIS SUBROUTINE SIMULATES AN AMPLITUDE-MODULATION (AM) DEMODU-
C LATOR.
C THE INPUT PARAMETERS ARE
C       NFC = NUMBER OF FREQUENCY COMPONENTS,
C       VF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C               VOLTAGE SPECTRUM COMPONENTS OF THE DEMODULATOR
C               INPUT SIGNAL.
C THE OUTPUT PARAMETER IS
C       SF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C               SPECTRUM COMPONENTS OF THE DEMODULATED SIGNAL
C               ARE TO BE STORED.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE.
C DECLARATION STATEMENT
      DIMENSION   VF(64),SF(64)
C PRELIMINARY PROCESSING
   10 NO=NFC
      NT2=NO*2
C FOURIER-TRANSFORMS THE INPUT SIGNAL FROM FREQUENCY DOMAIN TO
C TIME DOMAIN.
   20 CALL DFT(NO,VF,1,SF)
C EXTRACTS THE AMPLITUDE INFORMATION AT EACH SAMPLING POINT IN
C THE TIME DOMAIN AND STORES THE RESULTS IN THE SF ARRAY.
   30 DO 31  J=2,NT2,2
```

```
      SF(J-1)=SQRT(SF(J-1)**2+SF(J)**2)
      SF(J)  =0.0
   31 CONTINUE
C FOURIER-TRANSFORMS THE DEMODULATED SIGNAL FROM TIME DOMAIN TO
C FREQUENCY DOMAIN.
   40 CALL DFT(N0,SF,-1,SF)
C ADJUSTS THE DEMODULATED-SIGNAL SPECTRUM IN SUCH A WAY THAT
C ITS UPPER HALF BECOMES ZERO.
   50 J2=NT2+2
      DO 51  J1=4,N0,2
        J2=J2-2
        SF(J1-1)=SF(J1-1)+SF(J2-1)
        SF(J1)  =SF(J1)  -SF(J2)
        SF(J2-1)=0.0
        SF(J2)  =0.0
   51 CONTINUE
      RETURN
      END


      SUBROUTINE  DEMFM (NFC,UFS,VF,FCD,SF)
C THIS SUBROUTINE SIMULATES A FREQUENCY-MODULATION (FM) DEMODU-
C LATOR.
C THE INPUT PARAMETERS ARE
C      NFC = NUMBER OF FREQUENCY COMPONENTS,
C      UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C      VF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C            VOLTAGE SPECTRUM COMPONENTS OF THE DEMODULATOR
C            INPUT SIGNAL,
C      FCD = CENTER FREQUENCY OF THE DEMODULATOR.
C THE OUTPUT PARAMETER IS
C      SF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C            SPECTRUM COMPONENTS OF THE DEMODULATED SIGNAL
C            ARE TO BE STORED.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE.
C DECLARATION STATEMENTS
      DIMENSION   VF(64),SF(64)
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      F0=U0*AINT(FCD/U0+0.5)
      NT2=N0*2
C DIFFERENTIATES THE INPUT SIGNAL IN THE FREQUENCY DOMAIN AND
C STORES THE RESULT IN THE SF ARRAY.
   20 FI=-U0
      DO 21  J=2,NT2,2
        FI=FI+U0
        SF(J-1)=-FI*VF(J)
        SF(J)  = FI*VF(J-1)
   21 CONTINUE
C FOURIER-TRANSFORMS THE SF AND VF ARRAYS FROM FREQUENCY DOMAIN
C TO TIME DOMAIN.
   30 CALL DFT(N0,SF,1,SF)
      CALL DFT(N0,VF,1,VF)
C CALCULATES THE INSTANTANEOUS FREQUENCY AT EACH SAMPLING POINT
C IN THE TIME DOMAIN AND STORES THE RESULTS IN THE SF ARRAY.
   40 DO 41  J=2,NT2,2
        A=SF(J-1)
        B=SF(J)
        C=VF(J-1)
        D=VF(J)
        SF(J-1)=((B*C-A*D)/(C*C+D*D)-F0)
        SF(J)  =0.0
```

```
   41 CONTINUE
C FOURIER-TRANSFORMS THE DEMODULATED SIGNAL FROM TIME DOMAIN TO
C FREQUENCY DOMAIN.
   50 CALL DFT(N0,SF,-1,SF)
C RESTORES THE INPUT VF ARRAY.
      CALL DFT(N0,VF,-1,VF)
C ADJUSTS THE DEMODULATED-SIGNAL SPECTRUM IN SUCH A WAY THAT
C ITS UPPER HALF BECOMES ZERO.
   60 J2=NT2+2
      DO 61  J1=4,N0,2
        J2=J2-2
        SF(J1-1)=SF(J1-1)+SF(J2-1)
        SF(J1)  =SF(J1)  -SF(J2)
        SF(J2-1)=0.0
        SF(J2)  =0.0
   61 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  DEMFSK(NFC,UFS,VF,FCD,NIB,IB)
C THIS SUBROUTINE SIMULATES A FREQUENCY-SHIFT-KEYING (FSK) DE-
C MODULATOR.
C THE INPUT PARAMETERS ARE
C      NFC = NUMBER OF FREQUENCY COMPONENTS
C            (MUST NOT EXCEED 2048),
C      UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C      VF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C            VOLTAGE SPECTRUM COMPONENTS OF THE DEMODULATOR
C            INPUT SIGNAL,
C      FCD = CENTER FREQUENCY OF THE DEMODULATOR,
C      NIB = NUMBER OF INFORMATION BITS IN A PERIOD.
C THE OUTPUT PARAMETER IS
C      IB  = ARRAY OF DIMENSION NIB WHERE THE DEMODULATED
C            INFORMATION BITS ARE TO BE STORED.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE.
C DECLARATION STATEMENTS
      DIMENSION   VF(64),IB(16)
      DIMENSION   WK(4096)
      DATA  NFCMX/2048/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 N0=NFC
      IF(N0.GT.NFCMX)     GO TO 90
      U0=UFS
      F0=U0*AINT(FCD/U0+0.5)
      NI=NIB
      NT2=N0*2
      NI2=NI*2
C DIFFERENTIATES THE INPUT SIGNAL IN THE FREQUENCY DOMAIN AND
C STORES THE RESULT IN THE WK ARRAY.
   20 FI=-U0
      DO 21  J=2,NT2,2
        FI=FI+U0
        WK(J-1)=-FI*VF(J)
        WK(J)  = FI*VF(J-1)
   21 CONTINUE
C FOURIER-TRANSFORMS THE VF AND WK ARRAYS FROM FREQUENCY DOMAIN
C TO TIME DOMAIN.
   30 CALL DFT(N0,VF,1,VF)
      CALL DFT(N0,WK,1,WK)
C CALCULATES THE INSTANTANEOUS FREQUENCY AT EACH SAMPLING POINT
C IN THE TIME DOMAIN AND STORES THE RESULTS IN THE WK ARRAY.
```

```
   40 DO 41  I=1,N0
         J=I+I
         A=WK(J-1)
         B=WK(J)
         C=VF(J-1)
         D=VF(J)
         WK(I)=(B*C-A*D)/(C*C+D*D)-F0
   41 CONTINUE
C RESTORES THE INPUT VF ARRAY.
   50 CALL DFT(N0,VF,-1,VF)
C SIMULATES BIT SYNCHRONIZATION.
   60 NS=N0/NI2+1
      SMMX=0.0
      DO 64  IS=1,NS
         J=N0+IS
         WK(J)=WK(IS)
         SM=0.0
         IIBPV=1
         J=IS
         SB=WK(J)
         DO 63  I=2,N0
            IIB=((I-1)*NI)/N0+1
            IF(IIB.EQ.IIBPV)      GO TO 62
   61       SM=SM+ABS(SB)
            IIBPV=IIB
            SB=0.0
   62       J=J+1
            SB=SB+WK(J)
   63    CONTINUE
         SM=SM+ABS(SB)
         IF(SM.LE.SMMX)     GO TO 64
         SMMX=SM
         ISMX=IS
   64 CONTINUE
C ESTIMATES THE TRANSMITTED BIT SEQUENCE AND STORES THE RESULTS
C IN THE IB ARRAY.
   65 IIBPV=1
      J=ISMX
      SB=WK(J)
      DO 68  I=2,N0
         IIB=((I-1)*NI)/N0+1
         IF(IIB.EQ.IIBPV)   GO TO 67
   66    IB(IIBPV)=0
         IF(SB.GT.0.0)      IB(IIBPV)=1
         IIBPV=IIB
         SB=0.0
   67    J=J+1
         SB=SB+WK(J)
   68 CONTINUE
      IB(IIB)=0
      IF(SB.GT.0.0)        IB(IIB)=1
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) N0
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***    IMPROPER INPUT VALUE(S)./
     1    8H    NFC =,I6/
     2    35H ERROR DETECTED IN ROUTINE    DEMFSK/)
      END
```

```
      SUBROUTINE  DEMPHM(NFC,UFS,VF,SF)
C THIS SUBROUTINE SIMULATES A PHASE-MODULATION DEMODULATOR.
C THE INPUT PARAMETERS ARE
C      NFC = NUMBER OF FREQUENCY COMPONENTS,
C      UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C      VF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C            VOLTAGE SPECTRUM COMPONENTS OF THE DEMODULATOR
C            INPUT SIGNAL.
C THE OUTPUT PARAMETER IS
C      SF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C            SPECTRUM COMPONENTS OF THE DEMODULATED SIGNAL
C            ARE TO BE STORED.
C THE OUTPUT, SF, IS NORMALIZED IN SUCH A WAY THAT ITS POWER
C WILL EQUAL UNITY WHEN THE INPUT SIGNAL IS PHASE MODULATED BY
C A SINGLE SINUSOID WITH A MODULATION INDEX OF UNITY.
C THE CF CONSTANT IN THE DATA STATEMENT IS SQRT(2.0)*TWOPI,
C AND THE CH IS ONE HALF OF CF.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE.
C DECLARATION STATEMENTS
      DIMENSION   VF(64),SF(64)
      DATA  TWOPI/6.28318530717959/, CF/8.88576587631673/,
     1      CH/4.44288293815837/
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      NT2=N0*2
      FLN2=NT2
C FOURIER-TRANSFORMS THE INPUT SIGNAL FROM FREQUENCY DOMAIN TO
C TIME DOMAIN.
   30 CALL DFT(N0,VF,1,SF)
C EXTRACTS THE PHASE INFORMATION AT EACH SAMPLING POINT IN
C THE TIME DOMAIN AND STORES THE RESULTS IN THE SF ARRAY.
   40 DO 42  J=2,NT2,2
        CYCL=0.0
        A=SF(J-1)
        B=SF(J)
        IF(A.NE.0.0.OR.B.NE.0.0)    CYCL=ATAN2(B,A)/TWOPI
        IF(J.EQ.2)         GO TO 41
        DCYCL=CYCL-CYCLPV
        IF(DCYCL.LT.0.0)  CYCL=CYCL+AINT(-DCYCL+1.0)
   41   SF(J-1)=CYCL
        SF(J)  =0.0
        CYCLPV =CYCL
   42 CONTINUE
   45 DCYCL=AINT(SF(NT2-1)-SF(1)+1.0)
      DCYCL=DCYCL/FLN2
      DO 46  J=2,NT2,2
        SF(J-1)=CF*(SF(J-1)-DCYCL*FLOAT(J-2))
   46 CONTINUE
C FOURIER-TRANSFORMS THE DEMODULATED SIGNAL FROM TIME DOMAIN TO
C FREQUENCY DOMAIN.
   50 CALL DFT(N0,SF,-1,SF)
C ADJUSTS THE DEMODULATED-SIGNAL SPECTRUM IN SUCH A WAY THAT
C ITS UPPER HALF BECOMES ZERO.
   60 J2=NT2+2
      DO 61  J1=4,N0,2
        J2=J2-2
        SF(J1-1)=SF(J1-1)+SF(J2-1)
        SF(J1)  =SF(J1)  -SF(J2)
        SF(J2-1)=0.0
        SF(J2)  =0.0
   61 CONTINUE
   70 IF(SF(1).GE.-CH)    GO TO 71
      SF(1)=SF(1)+CF
```

```
      RETURN
   71 IF(SF(1).LE.CH)      RETURN
      SF(1)=SF(1)-CF
      RETURN
      END


      SUBROUTINE  DEMSSB(NFC,UFS,VF,FRD,SF)
C THIS SUBROUTINE SIMULATES A SINGLE-SIDEBAND (SSB) DEMODULATOR.
C THE INPUT PARAMETERS ARE
C     NFC = NUMBER OF FREQUENCY COMPONENTS.
C     UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM.
C     VF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C           VOLTAGE SPECTRUM COMPONENTS OF THE DEMODULATOR
C           INPUT SIGNAL.
C     FRD = REFERENCE FREQUENCY OF THE DEMODULATOR.
C THE OUTPUT PARAMETER IS
C     SF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C           SPECTRUM COMPONENTS OF THE DEMODULATED SIGNAL
C           ARE TO BE STORED.
C THIS SUBROUTINE CALLS THE RANF FUNCTION.
C DECLARATION STATEMENTS
      DIMENSION   VF(64),SF(64)
      DATA  TWOPI/6.283185307179S9/
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      F0=FRD
      NT2=2*N0
      IFR=F0/U0+1.5
      IFRT2=IFR*2
C GENERATES THE CARRIER LOCALLY.
   20 PHI=TWOPI*(RANF(0)-0.5)
      C1=COS(PHI)
      C2=SIN(PHI)
C TRANSLATES EACH SPECTRAL COMPONENT OF THE INPUT SIGNAL DOWN
C TO THE BASEBAND AND STORES THE RESULTS IN THE SF ARRAY.
   30 J1=IFRT2+2
      J2=IFRT2-2
      DO 34  K=2,NT2,2
        S1=0.0
        S2=0.0
        J1=J1-2
        J2=J2+2
        IF(J1.GT.NT2)      GO TO 33
        IF(K.EQ.2)         GO TO 32
        IF(J1.LE.0)        GO TO 31
        V1=VF(J1-1)
        V2=VF(J1)
        S1= C1*V1+C2*V2
        S2=-C1*V2+C2*V1
   31   IF(J2.GT.NT2)      GO TO 33
   32   V1=VF(J2-1)
        V2=VF(J2)
        S1=S1+C1*V1+C2*V2
        S2=S2+C1*V2-C2*V1
   33   SF(K-1)=S1
        SF(K)  =S2
   34 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  DFT(N,X1,ISGN,X2)
C THIS SUBROUTINE PERFORMS DISCRETE FOURIER TRANSFORM.  THIS IS
C A MODIFIED VERSION FROM THE DISCRETE COMPLEX FAST FOURIER
C TRANSFORM PACKAGE PROGRAMMED BY L. DAVID LEWIS AND MARIE WEST
C OF THE U.S. DEPARTMENT OF COMMERCE BOULDER LABORATORIES.  (CF.
C SINGLETON, COMMUN. ACM, VOL. 11, PP. 776-779, 1968, AND VOL.
C 12, P. 187, 1969)
C THE INPUT PARAMETERS ARE
C      N    = DIMENSION OF THE X1 AND X2 ARRAYS
C           = 2**M  (M=0,1,2,...,14),
C      X1   = COMPLEX ARRAY OF DIMENSION N CONTAINING THE
C             INPUT SEQUENCE TO BE TRANSFORMED,
C      ISGN = SIGN IN THE EXPONENT
C           = +1  FROM FREQUENCY SPECTRUM TO TIME SEQUENCE
C           = -1  FROM TIME SEQUENCE TO FREQUENCY SPECTRUM.
C THE OUTPUT PARAMETER IS
C      X2   = COMPLEX ARRAY OF DIMENSION N WHERE THE
C             TRANSFORMED SEQUENCE IS TO BE STORED.
C THE P2 ARRAY CONTAINS POWERS OF TWO  --  P2(I)=2**(I-1).
C THE ST ARRAY IS SINE ARRAY  --  ST(I)=SIN(PI/(2**I)).
C DECLARATION STATEMENTS
      DIMENSION   X1(2048),X2(2048)
      DIMENSION   P2(15),ST(15)
      INTEGER     P2
      REAL        IM
      DATA  P2(1),P2(2),P2(3),P2(4),P2(5),P2(6),P2(7),P2(8),
     1      P2(9),P2(10),P2(11),P2(12),P2(13),P2(14),P2(15)/
     2      1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,
     3      16384/
      DATA  ST(1),ST(2),ST(3),ST(4),ST(5),ST(6),ST(7),ST(8),
     1      ST(9),ST(10),ST(11),ST(12),ST(13),ST(14),ST(15)/
     2       .100000000000000E+1, .707106781186548E+0,
     3       .382683432365090E+0, .195090322016128E+0,
     4       .980171403295606E-1, .490676743274180E-1,
     5       .245412285229123E-1, .122715382857199E-1,
     6       .613588464915448E-2, .306795676296598E-2,
     7       .153398018628477E-2, .766990318742705E-3,
     7       .383495187571396E-3, .191747597310703E-3,
     8       .958737990959773E-4/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 N0=N
      DO 11  I=1,15
        IF(N0.EQ.P2(I))   GO TO 12
   11 CONTINUE
      GO TO 90
   12 M0=I-1
      NT2=N0*2
      DO 13  K=1,NT2
        X2(K)=X1(K)
   13 CONTINUE
      IF(M0.EQ.0)    RETURN
      NP1=N0+1
      NM1=N0-1
      NQ=N0/4
      ISGN0=ISGN
C REVERSIBLE PERMUTATION OF THE X2 ARRAY FROM NORMAL SEQUENCE
C TO REVERSE BINARY SEQUENCE.
   20 IF(M0.EQ.1)    GO TO 30
      I=M0
      K0=1
      KI=2
      KJ=NM1
   21 K0=K0+P2(I)
```

```
      P2(I)=-P2(I)
      IF(P2(I).GE.0)       GO TO 23
      IF(KI.GT.K0.OR.KJ.LT.K0)       GO TO 22
      KIT2=KI*2
      K0T2=KJ*2
      RE=X2(KIT2-1)
      IM=X2(KIT2)
      X2(KIT2-1)=X2(K0T2-1)
      X2(KIT2)  =X2(K0T2)
      X2(K0T2-1)=RE
      X2(K0T2)  =IM
      IF(KJ.EQ.K0)   GO TO 22
      KK=NP1-KJ
      KKT2=KK*2
      KJT2=KJ*2
      RE=X2(KKT2-1)
      IM=X2(KKT2)
      X2(KKT2-1)=X2(KJT2-1)
      X2(KKT2)  =X2(KJT2)
      X2(KJT2-1)=RE
      X2(KJT2)  =IM
   22 I=M0
      KI=KI+1
      KJ=KJ-1
      GO TO 21
   23 IF(I.EQ.2)      GO TO 30
      I=I-1
      GO TO 21
C FOURIER TRANSFORM FROM INPUT IN REVERSE BINARY SEQUENCE TO
C OUTPUT IN NORMAL SEQUENCE.
   30 DO 31  KK=1,N0,2
         KKT2=KK*2
         RE=X2(KKT2-1)-X2(KKT2+1)
         IM=X2(KKT2)   -X2(KKT2+2)
         X2(KKT2-1)=X2(KKT2-1)+X2(KKT2+1)
         X2(KKT2)  =X2(KKT2)   +X2(KKT2+2)
         X2(KKT2+1)=RE
         X2(KKT2+2)=IM
   31 CONTINUE
   40 IF(M0.EQ.1)     GO TO 50
      EXPS=1.0
      IF(ISGN0.LT.0)       EXPS=-1.0
      JSPAN=1
      K=NQ
      DO 43  I=2,M0
         SD=-ST(I-1)
         CD=2.0*ST(I)*ST(I)
         R=-2.0*CD
         CN=1.0
         CM=0.0
         SN=0.0
         SM=EXPS
         JSPANH=JSPAN
         JSPAN =JSPAN+JSPAN
         JJ=0
         KK=1
   41    KS=KK+JSPAN
         KKT2=KK*2
         KST2=KS*2
         RE=CN*X2(KST2-1)-SN*X2(KST2)
         IM=SN*X2(KST2-1)+CN*X2(KST2)
         X2(KST2-1)=X2(KKT2-1)-RE
         X2(KST2)  =X2(KKT2)   -IM
         X2(KKT2-1)=X2(KKT2-1)+RE
```

```
      X2(KKT2)  =X2(KKT2)  +IM
      KK=KK+JSPANH
      KS=KS+JSPANH
      KKT2=KK*2
      KST2=KS*2
      RE=CM*X2(KST2-1)-SM*X2(KST2)
      IM=SM*X2(KST2-1)+CM*X2(KST2)
      X2(KST2-1)=X2(KKT2-1)-RE
      X2(KST2)  =X2(KKT2)  -IM
      X2(KKT2-1)=X2(KKT2-1)+RE
      X2(KKT2)  =X2(KKT2)  +IM
      KK=KS+JSPANH
      IF(KK.LT.NO)       GO TO 41
      KK=KK-NM1
      JJ=JJ+K
      IF(JJ.GE.NQ)       GO TO 42
      CD=R*CN+CD
      CN=CD+CN
      SM=CN*EXPS
      SD=R*CM+SD
      CM=SD+CM
      SN=-CM*EXPS
      GO TO 41
   42 K=K/2
   43 CONTINUE
C DIVIDES EACH ELEMENT OF X2 BY N WHEN TRANSFORM IS FROM TIME
C SEQUENCE TO FREQUENCY SPECTRUM.
   50 IF(ISGNO.GT.0)       RETURN
      CF=1.0/FLOAT(NO)
      DO 51  K=1,NT2
        X2(K)=CF*X2(K)
   51 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) NO
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/24H ***   IMPROPER N VALUE./6H   N =,I5/
     1   35H ERROR DETECTED IN ROUTINE   DFT   /)
      END


      SUBROUTINE  FRBPFC(NOB,RPB,FCB,BWB,UFS,IFCMN,IFCMX,HFB)
C THIS SUBROUTINE CALCULATES FREQUENCY RESPONSE OF A BANDPASS
C FILTER (BPF) OF CHEBYSHEV TYPE.  (CF. STORER, PASSIVE NETWORK
C SYNTHESIS, MCGRAW-HILL, 1957)
C THE INPUT PARAMETERS ARE
C     NOB   = ORDER OF THE BPF
C             (MUST NOT BE NEGATIVE NOR GREATER THAN 100),
C     RPB   = RIPPLE (DB) WITHIN THE PASSBAND OF THE BPF
C             (MUST BE NONNEGATIVE),
C     FCB   = CENTER FREQUENCY OF THE BPF
C             (MUST BE POSITIVE),
C     BWB   = 3-DB BANDWIDTH OF THE BPF (MUST BE POSITIVE),
C     UFS   = UNIT FREQUENCY SPACING (MUST BE POSITIVE),
C     IFCMN = MIN. OF FREQ. COMPONENT NUMBER,
C     IFCMX = MAX. OF FREQ. COMPONENT NUMBER.
C THE OUTPUT PARAMETER IS
C     HFB   = COMPLEX ARRAY FOR THE FREQUENCY RESPONSE OF
C             THE BPF.
C DECLARATION STATEMENTS
      COMPLEX     HFB(100)
      COMPLEX     POLE(100),HFBI,HFC,NMRTR,Z
```

```
      DATA  NOBMX/100/
      DATA  NOPV/-1/, RPPV/-1.0/
      DATA  HLFPI/1.57079 63267/
      DATA  DRMN/1.0E-12/
      DATA  LUN/6/
      ASINH(X)=ALOG(X+SQRT(X*X+1.0))
C PRELIMINARY PROCESSING
   10 NO=NOB
      RP=RPB
      FC=FCB
      BW=BWB
      UF=UFS
      IMN=IFCMN
      IMX=IFCMX
      IF(NO.LT.0.OR.NO.GT.NOBMX)     GO TO 90
      IF(RP.LT.0.0)          GO TO 90
      IF(FC.LE.0.0)          GO TO 90
      IF(BW.LE.0.0)          GO TO 90
      IF(UF.LE.0.0)          GO TO 90
      IF(NO.EQ.0)            GO TO 70
C CALCULATES BASIC PARAMETERS WHEN NECESSARY.
   20 IF(NO.EQ.NOPV.AND.RP.EQ.RPPV)       GO TO 60
C - NOB.GE.3.AND.RPB.GT.0.0
      NOPV=NO
      RPPV=RP
      NOP1=NO+1
      FLNO=NO
      R=1.0
      HFC=1.0
      PHI0=HLFPI/FLNO
      IF(RP.EQ.0.0.OR.NO.EQ.1)       GO TO 40
      VR=EXP(0.115129254 6*RP)
      RE=1.0/SQRT(VR*VR-1.0)
      U0=ASINH(RE)/FLNO
      EXU=EXP(U0)
      REXU=1.0/EXU
      CH=0.5*(EXU+REXU)
      SH=0.5*(EXU-REXU)
      IF(NO-NO/2*2.EQ.0)  HFC=HFC/VR
      IF(NO.EQ.2)     GO TO 30
   21 TWOR=R+R
      TM2=R
      TM1=TWOR*R-1.0
      DM2=1.0
      DM1=TWOR+TWOR
      DO 22  K=3,NO
        T=TWOR*TM1-TM2
        D=TM1+TM1+TWOR*DM1-DM2
        TM2=TM1
        TM1=T
        DM2=DM1
        DM1=D
   22 CONTINUE
      RPV=R
      R=R-(T-RE)/D
      IF(ABS(R-RPV).GT.DRMN)    GO TO 21
      GO TO 50
C - NOB.EQ.2
   30 R=SQRT(0.5*(1.0+RE))
      GO TO 50
C - NOB.EQ.1.OR.RPB.EQ.0.0
   40 CH=1.0
      SH=1.0
C CALCULATES POLES.
```

```
   50 NMRTR=HFC
      DO 51  K=1,NO
        PHI=FLOAT(K+K-NOP1)*PHI0
        POLE(K)=CMPLX(-SH*COS(PHI),CH*SIN(PHI))
        NMRTR=-NMRTR*POLE(K)
   51 CONTINUE
C CALCULATES FREQUENCY RESPONSE.
   60 CF=2.0*R/BW
      DO 62  I=IMN,IMX
        Z=CMPLX(0.0,(FLOAT(I-1)*UF-FC)*CF)
        HFBI=NMRTR
        DO 61  K=1,NC
          HFBI=HFBI/(Z-POLE(K))
   61   CONTINUE
        HFB(I)=HFBI
   62 CONTINUE
      RETURN
C ALL PASS FILTER (NOB=0)
   70 DO 71  I=IMN,IMX
        HFB(I)=(1.0,0.0)
   71 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) NO,RP,FC,BW,UF
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***   IMPROPER INPUT VALUE(S)./
     1   3X,5HNOB =,I5,5X,5HRPB =,E11.3,5X,5HFCB =,E11.3,
     2   5X,5HBWB =,E11.3,5X,5HUFS =,E11.3/
     3   35H ERROR DETECTED IN ROUTICE   FRBPFC/)
      END


      SUBROUTINE  MODAM (NFC,UFS,SF,FC,FLMI,VF)
C THIS SUBROUTINE SIMULATES AN AMPLITUDE-MODULATION (AM) MODU-
C LATOR.
C THE INPUT PARAMETERS ARE
C      NFC  = NUMBER OF FREQUENCY COMPONENTS,
C      UFS  = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C      SF   = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C             VOLTAGE SPECTRUM COMPONENTS OF THE MODULATING
C             SIGNAL,
C      FC   = CENTER FREQUENCY OF THE MODULATED SIGNAL,
C      FLMI = MODULATION INDEX.
C THE OUTPUT PARAMETER IS
C      VF   = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C             SPECTRUM COMPONENTS OF THE MODULATED SIGNAL ARE
C             TO BE STORED.
C THIS SUBROUTINE CALLS THE RANF FUNCTION.
C DECLARATION STATEMENTS
      DIMENSION   SF(64),VF(64)
      REAL        MI
      DATA  SQRT2/1.41421356237310/,TWOPI/6.28318530717959/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 NO=NFC
      U0=UFS
      F0=FC
      MI=FLMI
      IF(NO.LT.4)     GO TO 90
      IF(U0.LE.0.0)   GO TO 90
      IFC=F0/U0+1.5
      IF(IFC.GE.NO)   GO TO 90
```

```
      IF(MI.GT.1.0)  GO TO 90
      NT2=NO*2
      IFCT2=IFC*2
C GENERATES THE CARRIER PHASE.
   20 PHI=TWOPI*(RANF(0)-0.5)
      CP=COS(PHI)
      SP=SIN(PHI)
C DETERMINES THE COEFFICIENTS FOR SIDEBAND SPECTRAL COMPONENTS.
   30 PS=0.0
      DO 31  J=3,NT2
        PS=PS+SF(J)**2
   31 CONTINUE
      PS=PS/2.0
      IF(PS.LE.0.0)  GO TO 91
      C=0.5*MI/SQRT(PS)
      C1=C*CP
      C2=C*SP
C CALCULATES EACH SPECTRAL COMPONENT OF THE MODULATED SIGNAL
C AND STORES THE RESULTS IN THE VF ARRAY.
C - DC COMPONENT.
   40 VF(1)=0.0
      VF(2)=0.0
C - LOWER SIDEBAND.
   42 KMX=IFCT2-2
      J=IFCT2
      DO 43  K=4,KMX,2
        J=J-2
        S1=SF(J-1)
        S2=SF(J)
        VF(K-1)= C1*S1+C2*S2
        VF(K)  =-C1*S2+C2*S1
   43 CONTINUE
C - CARRIER.
   44 VF(IFCT2-1)=CP*SQRT2
      VF(IFCT2)  =SP*SQRT2
C - UPPER SIDEBAND.
   46 KMN=IFCT2+2
      J=2
      DO 47  K=KMN,NT2,2
        J=J+2
        S1=SF(J-1)
        S2=SF(J)
        VF(K-1)= C1*S1-C2*S2
        VF(K)  = C1*S2+C2*S1
   47 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) NO,UO,FO
      GO TO 92
   91 WRITE (LUN,2091)
   92 WRITE (LUN,2092)
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***   IMPROPER INPUT VALUE(S)./
     1   8H   NFC =,I6,5X,5HUFS =,E11.3,5X,4HFC =,E11.3)
 2091 FORMAT(1X/33H ***   ALL ZERO COMPONENTS IN SF.)
 2092 FORMAT(35H ERROR DETECTED IN ROUTINE   MODAM /)
      END


      SUBROUTINE  MODFM (NFC,UFS,SF,FC,FDV,VF)
C THIS SUBROUTINE SIMULATES A FREQUENCY-MODULATION (FM) MODU-
C LATOR.
```

```
C THE INPUT PARAMETERS ARE
C      NFC = NUMBER OF FREQUENCY COMPONENTS,
C      UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C      SF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C            VOLTAGE SPECTRUM COMPONENTS OF THE MODULATING
C            SIGNAL,
C      FC  = CENTER FREQUENCY OF THE MODULATED SIGNAL,
C      FDV = FREQUENCY DEVIATION.
C THE OUTPUT PARAMETER IS
C      VF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C            SPECTRUM COMPONENTS OF THE MODULATED SIGNAL ARE
C            TO BE STORED.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE AND THE RANF
C FUNCTION.
C DECLARATION STATEMENTS
      DIMENSION   SF(64),VF(64)
      DATA  SQRT2/1.41421356237310/,TWOPI/6.28318530717959/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      F0=FC
      FD=FDV
      IF(N0.LT.4)     GO TO 90
      IF(U0.LE.0.0)   GO TO 90
      IFC=F0/U0+1.5
      IF(IFC.GE.N0)   GO TO 90
      F0=U0*FLOAT(IFC-1)
      NT2=N0*2
      DOMG=TWOPI*U0
C INTEGRATES THE MODULATING SIGNAL IN THE FREQUENCY DOMAIN,
C STORES THE RESULTS IN THE VF ARRAY, AND DETERMINES THE
C COEFFICIENTS FOR PHASE MODULATION.
   30 VF(1)=0.0
      VF(2)=0.0
      PS=0.0
      OMG=0.0
      DO 31  J=4,NT2,2
        OMG=OMG+DOMG
        VF(J-1)= SF(J)  /OMG
        VF(J)  =-SF(J-1)/OMG
        PS=PS+SF(J-1)**2+SF(J)**2
   31 CONTINUE
      IF(PS.LE.0.0)   GO TO 91
      C=FD/SQRT(PS)
C FOURIER-TRANSFORMS THE VF ARRAY FROM FREQUENCY DOMAIN TO TIME
C DOMAIN.
   40 CALL DFT(N0,VF,1,VF)
C PHASE-MODULATES THE CARRIER, AT EACH SAMPLING POINT IN THE
C TIME DOMAIN, WITH THE VF ARRAY AND STORES THE RESULTS IN THE
C VF ARRAY.
   50 FMX=U0*FLOAT(N0)
      CC=RANF(0)
      FI=-F0
      DO 51  J=2,NT2,2
        FI=FI+F0
        CYCL=FI/FMX+CC
        CYCL=CYCL-AINT(CYCL+0.5)
        PHI=TWOPI*(CYCL+C*VF(J-1))
        VF(J-1)=SQRT2*COS(PHI)
        VF(J)  =SQRT2*SIN(PHI)
   51 CONTINUE
C FOURIER-TRANSFORMS THE MODULATED SIGNAL FROM TIME DOMAIN TO
C FREQUENCY DOMAIN.
```

```
   60 CALL DFT(N0,VF,-1,VF)
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) N0,U0,F0
      GO TO 92
   91 WRITE (LUN,2091)
   92 WRITE (LUN,2092)
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***    IMPROPER INPUT VALUE(S)./
     1    8H    NFC =,I6,5X,5HUFS =,E11.3,5X,4HFC =,E11.3)
 2091 FORMAT(1X/33H ***    ALL ZERO COMPONENTS IN SF.)
 2092 FORMAT(35H ERROR DETECTED IN ROUTINE   MODFM /)
      END


      SUBROUTINE  MODFSK(NFC,UFS,NIB,IB,FC,FS,VF)
C THIS SUBROUTINE SIMULATES A FREQUENCY-SHIFT-KEYING (FSK) MODU-
C LATOR.
C THE INPUT PARAMETERS ARE
C      NFC = NUMBER OF FREQUENCY COMPONENTS,
C      UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C      NIB = NUMBER OF INFORMATION BITS IN A PERIOD
C            (MUST NOT EXCEED NFC/2),
C      IB  = ARRAY OF DIMENSION NIB CONTAINING THE INFORMA-
C            TION BITS,
C      FC  = CENTER FREQUENCY OF THE MODULATED SIGNAL,
C      FS  = FREQUENCY SHIFT.
C THE OUTPUT PARAMETER IS
C      VF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C            SPECTRUM COMPONENTS OF THE MODULATED SIGNAL ARE
C            TO BE STORED.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE AND THE RANF
C FUNCTION.
C DECLARATION STATEMENTS
      DIMENSION   IB(16),VF(64)
      DATA  SQRT2/1.41421356237310/, TWOPI/6.28318530717959/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      NI=NIB
      FC0=FC
      FS0=FS
      IF(N0.LT.4)     GO TO 90
      IF(U0.LE.0.0)   GO TO 90
      IF(2*NI.GT.N0)  GO TO 90
      F0=U0*AINT(FC0/U0+0.5)-0.5*FS0
      F1=F0+FS0
      FMX=U0*FLOAT(N0)
      IF(F0.LE.0.0)   GO TO 90
      IF(F1.GE.FMX)   GO TO 90
      FLNI=NI
      DLT=1.0/FMX
C FREQUENCY-SHIFT-KEYS THE CARRIER, AT EACH SAMPLING POINT IN
C THE TIME DOMAIN, BY CHANGING THE CARRIER PHASE ACCORDING TO
C THE INFORMATION BIT SEQUENCE AND STORES THE RESULTS IN THE
C VF ARRAY.
   50 CYCL=RANF(0)
      IBIPV=IB(1)
      DO 59  I=1,N0
        IF(I.EQ.1)          GO TO 58
        IIB=((I-1)*NI)/N0+1
```

```
      IBI=IB(IIB)
      IF(IBI.EQ.IBIPV)  GO TO 54
   51 IBIPV=IBI
      TT=FLOAT(NO*(IIB-1))/FLNI
      DLT1=(TT-FLOAT(I-2))*DLT
      DLT2=DLT-DLT1
      IF(IBI.GT.0)      GO TO 53
   52 CYCL=CYCL+F1*DLT1+F0*DLT2
      GO TO 58
   53 CYCL=CYCL+F0*DLT1+F1*DLT2
      GO TO 58
   54 IF(IBI.GT.0)      GO TO 56
   55 CYCL=CYCL+F0*DLT
      GO TO 58
   56 CYCL=CYCL+F1*DLT
   58 CYCL=CYCL-AINT(CYCL+0.5)
      PHI=TWOPI*CYCL
      J=I+I
      VF(J-1)=SQRT2*COS(PHI)
      VF(J)  =SQRT2*SIN(PHI)
   59 CONTINUE
C FOURIER-TRANSFORMS THE MODULATED SIGNAL FROM TIME DOMAIN TO
C FREQUENCY DOMAIN.
   60 CALL DFT(NO,VF,-1,VF)
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) NO,UO,NI,FC0,FS0
   92 WRITE (LUN,2092)
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***   IMPROPER INPUT VALUE(S)./
     1   8H   NFC =,I6,5X,5HUFS =,E11.3,5X,5HNIB =,I6,
     2   5X,5HFC  =,E11.3,5X,5HFS  =,E11.3)
 2092 FORMAT(35H ERROR DETECTED IN ROUTINE   MODFSK/)
      END



      SUBROUTINE  MOLPHM(NFC,UFS,SF,FC,FLMI,VF)
C THIS SUBROUTINE SIMULATES A PHASE-MODULATION MODULATOR.
C THE INPUT PARAMETERS ARE
C     NFC  = NUMBER OF FREQUENCY COMPONENTS,
C     UFS  = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C     SF   = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C            VOLTAGE SPECTRUM COMPONENTS OF THE MODULATING
C            SIGNAL,
C     FC   = CENTER FREQUENCY OF THE MODULATED SIGNAL,
C     FLMI = MODULATION INDEX (IN RADIANS).
C THE OUTPUT PARAMETER IS
C     VF   = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C            SPECTRUM COMPONENTS OF THE MODULATED SIGNAL ARE
C            TO BE STORED.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE AND THE RANF
C FUNCTION.
C DECLARATION STATEMENTS
      DIMENSION   SF(64),VF(64)
      REAL        MI
      DATA  SQRT2/1.41421356237310/, TWOPI/6.28318530717959/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 NO=NFC
      UO=UFS
      F0=FC
      MI=FLMI
```

```
      IF(NO.LT.4)     GO TO 90
      IF(UO.LE.0.0)   GO TO 90
      IFC=FO/UO+1.5
      IF(IFC.GE.NO)   GO TO 90
      FO=UO*FLOAT(IFC-1)
      NT2=NO*2
C DETERMINES THE COEFFICIENT FOR PHASE MODULATION.
   30 PS=0.0
      DO 31  J=1,NT2
        PS=PS+SF(J)**2
   31 CONTINUE
      IF(PS.LE.0.0)   GO TO 91
      C=MI/SQRT(PS)
C FOURIER-TRANSFORMS THE MODULATING SIGNAL FROM FREQUENCY DOMAIN
C TO TIME DOMAIN AND STORES THE RESULT IN THE VF ARRAY.
   40 CALL DFT(NO,SF,1,VF)
C PHASE-MODULATES THE CARRIER, AT EACH SAMPLING POINT IN THE
C TIME DOMAIN, WITH THE VF ARRAY AND STORES THE RESULTS IN
C THE VF ARRAY.
   50 FMX=UO*FLOAT(NO)
      CC=RANF(-1)
      FI=-FO
      DO 51  J=2,NT2,2
        FI=FI+FO
        CYCL=FI/FMX+CC
        CYCL=CYCL-AINT(CYCL+0.5)
        PHI=TWOPI*CYCL+C*VF(J-1)
        VF(J-1)=SQRT2*COS(PHI)
        VF(J)  =SQRT2*SIN(PHI)
   51 CONTINUE
C FOURIER-TRANSFORMS THE MODULATED SIGNAL FROM TIME DOMAIN TO
C FREQUENCY DOMAIN.
   60 CALL DFT(NO,VF,-1,VF)
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) NO,UO,FO
      GO TO 92
   91 WRITE (LUN,2091)
   92 WRITE (LUN,2092)
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***   IMPROPER INPUT VALUE(S)./
     1    8H   NFC =,I6,5X,5HUFS =,E11.3,5X,4HFC =,E11.3)
 2091 FORMAT(1X/33H ***   ALL ZERO COMPONENTS IN SF.)
 2092 FORMAT(35H ERROR DETECTED IN ROUTINE   MODPHM/)
      END


      SUBROUTINE  MODSSB(NFC,UFS,SF,FR,VF)
C THIS SUBROUTINE SIMULATES A SINGLE-SIDEBAND (SSB) MODULATOR.
C THE INPUT PARAMETERS ARE
C      NFC = NUMBER OF FREQUENCY COMPONENTS,
C      UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C      SF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C            VOLTAGE SPECTRUM COMPONENTS OF THE MODULATING
C            SIGNAL,
C      FR  = REFERENCE FREQUENCY OF THE MODULATED SIGNAL.
C THE OUTPUT PARAMETER IS
C      VF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C            SPECTRUM COMPONENTS OF THE MODULATED SIGNAL ARE
C            TO BE STORED.
C THIS SUBROUTINE CALLS THE RANF FUNCTION.
C DECLARATION STATEMENTS
```

```
      DIMENSION   SF(64),VF(64)
      DATA  TWOPI/6.28318530717959/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      F0=FR
      IF(N0.LT.4)     GO TO 90
      IF(U0.LE.0.0)   GO TO 90
      IFR=F0/U0+1.5
      IF(IFR.GE.N0)   GO TO 90
      NT2=N0*2
      IFRT2=IFR*2
C GENERATES THE CARRIER PHASE.
   20 PHI=TWOPI*(RANF(0)-0.5)
      CP=COS(PHI)
      SP=SIN(PHI)
C DETERMINES THE COEFFICIENTS FOR SSB MODULATION.
   30 PS=0.0
      DO 31  J=3,NT2
        PS=PS+SF(J)**2
   31 CONTINUE
      PS=PS/2.0
      IF(PS.LE.0.0)  GO TO 91
      C=1.0/SQRT(PS)
      C1=C*CP
      C2=C*SP
C TRANSLATES EACH COMPONENT OF THE MODULATING BASEBAND SIGNAL
C UP TO THE RF BAND AND STORES THE RESULTS IN THE VF ARRAY.
   40 DO 41  K=1,IFRT2
        VF(K)  = 0.0
   41 CONTINUE
   46 KMN=IFRT2+2
      J=2
      DO 47  K=KMN,NT2,2
        J=J+2
        S1=SF(J-1)
        S2=SF(J)
        VF(K-1)=C1*S1-C2*S2
        VF(K)  =C1*S2+C2*S1
   47 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) N0,U0,F0
      GO TO 92
   91 WRITE (LUN,2091)
   92 WRITE (LUN,2092)
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***   IMPROPER INPUT VALUE(S)./
     1   8H    NFC =,I6,5X,5HUFS =,E11.3,5X,4HFR =,E11.3)
 2091 FORMAT(1X/33H ***   ALL ZERO COMPONENTS IN SF.)
 2092 FORMAT(35H ERROR DETECTED IN ROUTINE   MODSSB/)
      END


      SUBROUTINE  PSPHCT(NFC,UFS,SF,PW)
C THIS SUBROUTINE SIMULATES A PSOPHOMETER FOR COMMERCIAL-
C TELEPHONE CIRCUIT.  (CF. CCITT RECOMMENDATION P.53)
C THE INPUT PARAMETERS ARE
C     NFC = NUMBER OF FREQUENCY COMPONENTS,
C     UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C     SF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
```

```
C           VOLTAGE SPECTRUM COMPONENTS OF THE INPUT SIGNAL.
C THE OUTPUT PARAMETER IS
C      PW  = PSOPHOMETRICALLY-WEIGHTED SIGNAL POWER.
C DECLARATION STATEMENTS
      DIMENSION   SF(64)
      DIMENSION   WT(500)
      DIMENSION   ALF(30),DB(30)
      DATA  ALF
     1  /1.22185, 1.69897, 2.00000, 2.17609, 2.30103,
     2    2.39794, 2.47712, 2.54407, 2.60206, 2.65321,
     3    2.69897, 2.77815, 2.84510, 2.90309, 2.95424,
     4    3.00000, 3.04139, 3.07918, 3.11394, 3.17609,
     5    3.30103, 3.34242, 3.39794, 3.47712, 3.54407,
     6    3.56820, 3.60206, 3.63347, 3.65321, 3.69897/
      DATA  DB
     1  / -85.00,  -63.00,  -41.00,  -29.00,  -21.00,
     2     -15.00,  -10.60,   -8.50,   -6.30,   -4.70,
     3      -3.60,   -2.00,   -0.90,    0.00,    0.60,
     4       1.00,    0.60,    0.00,   -0.40,   -1.30,
     5      -3.00,   -3.48,   -4.20,   -5.60,   -8.50,
     6     -10.70,  -15.00,  -20.70,  -25.00,  -36.00/
      DATA  NPV/0/, UPV/0.0/, FMAX/10000.0/, NMAX/500/
C PRELIMINARY PROCESSING
   10 N0=NFC-1
      U0=UFS
      FMX=U0*FLOAT(N0)
      IF(FMX.LE.FMAX)     GO TO 11
      FMX=FMAX
      N0=FMX/U0
   11 NG=1
   15 IF(N0.LE.NMAX)      GO TO 20
      N0=(N0-1)/3
      NG=NG*3
      U0=U0*3.0
      GO TO 15
C CALCULATES THE WEIGHTING COEFFICIENTS WHEN NECESSARY.
   20 IF(N0.LE.NPV.AND.U0.EQ.UPV)   GO TO 30
      NPV=N0
      UPV=U0
      JPV=0
      J=2
      FRI=0.0
      DO 29  I=1,N0
        FRI=FRI+U0
        IF(FRI.LE.0.0001)       GO TO 21
        ALFI=0.4342944819*ALOG(FRI)
        GO TO 22
   21   ALFI=-4.0
   22   IF(ALFI.LT.ALF(J))      GO TO 23
        IF(J.GE.30)  GO TO 23
        J=J+1
        GO TO 22
   23   IF(J.EQ.JPV)       GO TO 24
        JPV=J
        A0=ALF(J-1)
        C0=DB(J-1)
        C1=(DB(J)-DB(J-1))/(ALF(J)-ALF(J-1))
   24   WTI=C0+C1*(ALFI-A0)
        WT(I)=EXP(0.2302585093*WTI)
   29 CONTINUE
C CALCULATES THE WEIGHTED POWER.
   30 P0=0.0
      IFC=NG
      DO 39  I=1,N0
```

```
      DO 38  J=1,NG
        IFC=IFC+2
        P0=P0+WT(I)*(SF(IFC)**2+SF(IFC+1)**2)
 38   CONTINUE
 39 CONTINUE
    PW=0.5*P0*UFS
    RETURN
    END


    SUBROUTINE  PSPHPT(NFC,UFS,SF,PW)
C THIS SUBROUTINE SIMULATES A PSOPHOMETER FOR PROGRAM-
C TRANSMISSION CIRCUIT.  (CF. CCITT RECOMMENDATION P.53)
C THE INPUT PARAMETERS ARE
C     NFC = NUMBER OF FREQUENCY COMPONENTS,
C     UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C     SF  = COMPLEX ARRAY OF DIMENSION NFC CONTAINING THE
C           VOLTAGE SPECTRUM COMPONENTS OF THE INPUT SIGNAL.
C THE OUTPUT PARAMETER IS
C     PW  = PSOPHOMETRICALLY-WEIGHTED SIGNAL POWER.
C DECLARATION STATEMENTS
      DIMENSION   SF(64)
      DIMENSION   WT(500)
      DIMENSION   ALF(30),DB(30)
      DATA  ALF
     1  /1.00000, 1.30103, 1.39794, 1.47712, 1.60206,
     2   1.69897, 1.77815, 1.90309, 2.00000, 2.17609,
     3   2.30103, 2.47712, 2.60206, 2.77815, 2.90309,
     4   3.00000, 3.17609, 3.30103, 3.47712, 3.60206,
     5   3.69897, 3.77815, 3.84510, 3.90309, 3.95424,
     6   4.00000, 4.11394, 4.17609, 4.30103, 4.60206/
      DATA  DB
     1  / -41.50,  -41.50,  -41.00,  -40.00,  -37.50,
     2    -34.30,  -32.20,  -29.10,  -26.10,  -20.60,
     3    -17.30,  -12.70,   -8.60,   -4.80,   -1.90,
     4      0.00,    3.20,    5.30,    7.00,    8.20,
     5      8.40,    8.20,    7.30,    5.10,   -0.30,
     6     -9.70,  -30.00,  -33.00,  -36.00,  -40.00/
      DATA  NPV/0/, UPV/0.0/, FMAX/40000.0/, NMAX/500/
C PRELIMINARY PROCESSING
   10 N0=NFC-1
      U0=UFS
      FMX=U0*FLOAT(N0)
      IF(FMX.LE.FMAX)     GO TO 11
      FMX=FMAX
      N0=FMX/U0
   11 NG=1
   15 IF(NG.LE.NMAX)      GO TO 20
      N0=(N0-1)/3
      NG=NG*3
      U0=U0*3.0
      GO TO 15
C CALCULATES THE WEIGHTING COEFFICIENTS WHEN NECESSARY.
   20 IF(N0.LE.NPV.AND.U0.EQ.UPV)   GO TO 30
      NPV=N0
      UPV=U0
      JPV=0
      J=2
      FRI=0.0
      DO 29  I=1,N0
        FRI=FRI+U0
        IF(FRI.LE.0.0001)      GO TO 21
        ALFI=0.4342944819*ALOG(FRI)
```

```
      GO TO 22
   21 ALFI=-4.0
   22 IF(ALFI.LT.ALF(J))      GO TO 23
      IF(J.GE.30)  GO TO 23
      J=J+1
      GO TO 22
   23 IF(J.EQ.JPV)       GO TO 24
      JPV=J
      A0=ALF(J-1)
      C0=DB(J-1)
      C1=(DB(J)-DB(J-1))/(ALF(J)-ALF(J-1))
   24 WTI=C0+C1*(ALFI-A0)
      WT(I)=EXP(0.2302585093*WTI)
   29 CONTINUE
C CALCULATES THE WEIGHTED POWER.
   30 P0=0.0
      IFC=NG
      DO 39  I=1,N0
        DO 38  J=1,NG
          IFC=IFC+2
          P0=P0+WT(I)*(SF(IFC)**2+SF(IFC+1)**2)
   38   CONTINUE
   39 CONTINUE
      PW=0.5*P0*UFS
      RETURN
      END


      SUBROUTINE  SGPLS (NFC,UFS,FC,PRF,TR,TW,TF,DVF,VF)
C THIS SUBROUTINE SIMULATES A PULSE SIGNAL GENERATOR.
C THE INPUT PARAMETERS ARE
C     NFC = NUMBER OF FREQUENCY COMPONENTS,
C     UFS = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C     FC  = CENTER FREQUENCY OF OUTPUT SIGNAL,
C     PRF = PULSE REPETITION FREQUENCY,
C     TR  = PULSE RISE TIME,
C     TW  = PULSE WIDTH AT THE TOP,
C     TF  = PULSE FALL TIME,
C     DVF = MAXIMUM FREQUENCY DEVIATION.
C THE OUTPUT PARAMETER IS
C     VF  = COMPLEX ARRAY OF DIMENSION NFC WHERE THE VOLTAGE
C           SPECTRUM COMPONENTS OF THE OUTPUT SIGNAL ARE TO
C           BE STORED.
C THIS SUBROUTINE CALLS THE DFT SUBROUTINE AND THE RANF
C FUNCTION.
C DECLARATION STATEMENTS
      DIMENSION   VF(64)
      DATA  SQRT2/1.41421356237310/,TWOPI/6.28318530717959/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      F0=FC
      P0=PRF
      IF(N0.LT.4)          GO TO 90
      IF(U0.LE.0.0)        GO TO 90
      IF(P0.LE.0.0)        GO TO 90
      TR0=TR
      TW0=TW
      TF0=TF
      D0=DVF
      FMX=U0*FLOAT(N0)
      P0=U0*AINT(P0/U0+0.5)
```

```
      FB=F0-D0
      FE=F0+D0
      IF(FB.LE.0.0.OR.FB.GE.FMX)     GO TO 90
      IF(FE.LE.0.0.OR.FE.GE.FMX)     GO TO 90
      PRO=1.0/P0
      T2=(PRO-TW0)/2.0
      T1=T2-TR0
      T3=T2+TW0
      T4=T3+TF0
      IF(T1.LT.0.0)        GO TO 90
      IF(T4.GT.PRO)        GO TO 90
      T41=T4-T1
      IF(T41.LE.0.0)       GO TO 90
C DETERMINES THE COEFFICIENTS FOR PHASE MODULATION.
   20 C0=RANF(0)
      C1=(FB*T4-FE*T1)/T41
      C2=(FE-FB)/(2.0*T41)
C GENERATES THE PULSE SIGNAL AT EACH SAMPLING POINT IN THE TIME
C DOMAIN AND STORES THE RESULTS IN THE VF ARRAY.
   30 DO 39  I=1,N0
        J=I+I
        T=FLOAT(I-1)/FMX
        T0=AMOD(T,PRO)
        IF(T0.LE.T1)       GO TO 38
        IF(T0.GE.T4)       GO TO 38
        IF(T0.GT.T3)       GO TO 33
        IF(T0.GE.T2)       GO TO 32
   31   A=(T0-T1)/TR0
        GO TO 34
   32   A=1.0
        GO TO 34
   33   A=(T4-T0)/TF0
   34   A=SQRT2*A
        CYCL=C0+T*(C1+T*C2)
        CYCL=AMOD(CYCL+0.5,1.0)-0.5
        PHI=TWOPI*CYCL
        VF(J-1)=A*COS(PHI)
        VF(J)  =A*SIN(PHI)
        GO TO 39
   38   VF(J-1)=0.0
        VF(J)  =0.0
   39 CONTINUE
C FOURIER-TRANSFORMS THE GENERATED SIGNAL FROM TIME DOMAIN TO
C FREQUENCY DOMAIN.
   40 CALL DFT(N0,VF,-1,VF)
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) N0,U0,F0,P0,TR0,TW0,TF0,D0
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***   IMPROPER INPUT VALUE(S)./
     1   3X,5HNFC =,I6,   5X,5HUFS =,E11.3,5X,5HFC  =,E11.3,
     2   5X,5HPRF =,E11.3/3X,5HTR  =,E11.3,5X,5HTW  =,E11.3,
     3   5X,5HTF  =,E11.3,5X,5HDVF =,E11.3/
     4   35H ERROR DETECTED IN ROUTINE   SGPLS /)
      END


      SUBROUTINE  SGWGN (NFC,UFS,IFCMN,IFCMX,VF)
C THIS SUBROUTINE SIMULATES A WHITE GAUSSIAN NOISE GENERATOR.
C (CF. BOX AND MULLER, ANN. MATH. STAT., VOL. 28, PP. 610-613,
C 1958)
C THE INPUT PARAMETERS ARE
```

```
C       NFC   = NUMBER OF FREQUENCY COMPONENTS,
C       UFS   = UNIT FREQUENCY SPACING OF THE SPECTRUM,
C       IFCMN = MINIMUM OF FREQUENCY-COMPONENT NUMBER FOR
C               WHICH COMPONENTS OF VF ARE TO BE GENERATED
C               (MUST BE 2 OR GREATER),
C       IFCMX = MAXIMUM OF FREQUENCY-COMPONENT NUMBER FOR
C               WHICH COMPONENTS OF VF ARE TO BE GENERATED
C               (MUST BE NFC OR LESS).
C THE OUTPUT PARAMETER IS
C       VF    = COMPLEX ARRAY OF DIMENSION NFC WHERE THE
C               VOLTAGE SPECTRUM COMPONENTS OF THE OUTPUT
C               SIGNAL ARE TO BE STORED.
C THIS SUBROUTINE CALLS THE RANF FUNCTION.
C DECLARATION STATEMENTS
      DIMENSION   VF(64)
      DATA  TWOPI/6.2831853071795 9/
      DATA  LUN/6/
C PRELIMINARY PROCESSING
   10 N0=NFC
      U0=UFS
      IMN=IFCMN
      IMX=IFCMX
      IF(N0.LT.4)           GO TO 90
      IF(U0.LE.0.0)         GO TO 90
      IF(IMN.LT.2)          GO TO 90
      IF(IMX.GT.N0)         GO TO 90
      IF(IMN.GT.IMX)        GO TO 90
C CLEARS THE VF ARRAY.
   30 NT2=N0+N0
      DO 31  J=1,NT2
        VF(J)=0.0
   31 CONTINUE
C GENERATES A PAIR OF NORMAL DEVIATES AS THE REAL AND IMAGINARY
C PARTS OF EACH SPECTRAL COMPONENT OF GAUSSIAN NOISE AND STORES
C THE RESULTS IN THE VF ARRAY.
   40 CF=2.0*U0
      DO 41  I=IMN,IMX
        PP=RANF(0)
        A=SQRT(-CF*ALOG(PP))
        PH=TWOPI*RANF(0)
        J=I+I
        VF(J-1)=A*COS(PH)
        VF(J)  =A*SIN(PH)
   41 CONTINUE
      RETURN
C ERROR EXIT
   90 WRITE (LUN,2090) N0,U0,IMN,IMX
      RETURN
C FORMAT STATEMENTS FOR ERROR MESSAGES
 2090 FORMAT(1X/31H ***    IMPROPER INPUT VALUE(S)./
     1    3X,5HNFC =,I6,    5X,5HUFS =,E11.3,
     2    5X,7HIFCMN =,I6,5X,7HIFCMX =,I6/
     3    35H ERROR DETECTED IN ROUTINE    SGWGN /)
      END
```

APPENDIX B

USER MANUAL OF THE MODEL

Names and brief descriptions of all subprograms included in the model are listed in table B-1, which is a reproduction of table 1 given in section 4 of the text. Table B-1 is followed by user write-ups of all the subprograms in alphabetical order.

Table B-1. List of Subroutines Included in the Model

Name	Brief Description of Action
BPFCH	Simulates a bandpass filter of Chebyshev type.
CBCCT2	Simulates an RF combining circuit that linearly combines two RF signals.
CBCCT3	Simulates an RF combining circuit that linearly combines three RF signals.
DEMAM	Simulates an AM demodulator.
DEMFM	Simulates an FM demodulator.
DEMFSK	Simulates an FSK demodulator.
DEMPHM	Simulates a phase-modulation demodulator.
DEMSSB	Simulates an SSB demodulator.
DFT	Performs discrete Fourier transform.
FRBPFC	Calculates frequency response of a bandpass filter of Chebyshev type.
MODAM	Simulates an AM modulator.
MODFM	Simulates an FM modulator.
MODFSK	Simulates an FSK modulator.
MODPHM	Simulates a phase-modulation modulator.
MODSSB	Simulates an SSB modulator.
PSPHCT	Simulates a psophometer for commercial-telephone circuit.
PSPHPT	Simulates a psophometer for program-transmission circuit.
SGPLS	Simulates a pulse-signal generator.
SGWGN	Simulates a white-Gaussian-Noise generator.

Purpose: This subroutine simulates a bandpass filter of Chebyshev type.

Fortran Calling Statement:

```
CALL BPFCH (NFC,UFS,NØB,RPB,FCB,BWB,VF1,VF2)
```

where the input parameters are

NFC = number of frequency components (must be positive but not greater than 2048),

UFS = unit frequency spacing (must be positive),

NØB = order of the bandpass filter (must not be negative or greater than 100),

RPB = ripple (dB) within the passband of the bandpass filter (must be nonnegative),

FCB = center frequency of the bandpass filter (must be positive),

BWB = 3-dB bandwidth of the bandpass filter (must be positive),

VF1 = complex array of dimension NFC containing the spectrum components of the input voltage,

and the output parameter is

VF2 = complex array of dimension NFC where the spectrum components of the output voltage are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the FRBPFC subroutine.

Purpose: This subroutine simulates a radio-frequency (RF) combining circuit that linearly combines two RF signals with a specified ratio. The level of the first input signal is unchanged in this circuit.

Fortran Calling Statement:

```
CALL CBCCT2 (NFC,VF1,VF2,RDB,VF3)
```

where the input parameters are

NFC = number of frequency components in the spectrum of each signal,

VF1 = complex array of dimension NFC containing the voltage spectrum components of the first input signal,

VF2 = complex array of dimension NFC containing the voltage spectrum components of the second input signal,

RDB = signal-to-interference ratio or signal-to-noise-density ratio between the first and the second input signals in dB,

and the output parameter is

VF3 = complex array of dimension NFC where the voltage spectrum components of the composite signal are to be stored.

Error Message: None.

Purpose: This subroutine simulates a radio-frequency (RF) combining circuit that linearly combines three RF signals with specified ratios. The level of the first input signal is unchanged in this circuit.

Fortran Calling Statement:

```
CALL CBCCT3 (NFC,VF1,VF2,VF3,R12DB,R13DB,VF4)
```

where the input parameters are

- NFC = number of frequency components in the spectrum of each signal,
- VF1 = complex array of dimension NFC containing the voltage spectrum components of the first input signal,
- VF2 = complex array of dimension NFC containing the voltage spectrum components of the second input signal,
- VF3 = complex array of dimension NFC containing the voltage spectrum components of the third input signal,
- R12DB = signal-to-interference ratio or signal-to-noise-density ratio between the first and the second input signals in dB,
- R13DB = signal-to-interference ratio or signal-to-noise-density ratio between the first and the third input signals in dB,

and the output parameter is

- VF4 = complex array of dimension NFC where the voltage spectrum components of the composite signal are to be stored.

Error Message: None.

Purpose: This subroutine simulates an amplitude-modulation (AM) demodulator.

Fortran Calling Statement:

```
CALL DEMAM (NFC,VF,SF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

VF = complex array of dimension NFC containing the voltage spectrum components of the demodulator input signal,

and the output parameter is

SF = complex array of dimension NFC where the voltage spectrum components of the demodulated signal are to be stored.

Error Message: None.

Note: This subroutine calls the DFT subroutine.

Purpose: This subroutine simulates a frequency-modulation (FM) demodulator.

Fortran Calling Statement:

```
CALL DEMFM (NFC,UFS,VF,FCD,SF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum,

VF = complex array of dimension NFC containing the voltage spectrum components of the demodulator input signal,

FCD = center frequency of the demodulator,

and the output parameter is

SF = complex array of dimension NFC where the voltage spectrum components of the demodulated signal are to be stored.

Error Message: None.

Note: This subroutine calls the DFT subroutine.

Purpose: This subroutine simulates a frequency-shift-keying (FSK) demodulator.

Fortran Calling Statement:

```
CALL DEMFSK (NFC,UFS,VF,FCD,NIB,IB)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,11),

UFS = unit frequency spacing of the spectrum,

VF = complex array of dimension NFC containing the voltage spectrum components of the demodulator input signal,

FCD = center frequency of the demodulator,

NIB = number of information bits in a period,

and the output parameter is

IB = array of dimension NIB where the demodulated information bits are to be stored.

Error Message: When NFC exceed 2048 (=2**11), an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the DFT subroutine.

Purpose: This subroutine simulates a phase-modulation demodulator.

Fortran Calling Statement:

```
CALL DEMPHM (NFC,UFS,VF,SF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum,

VF = complex array of dimension NFC containing the voltage spectrum components of the demodulator input signal,

and the output parameter is

SF = complex array of dimension NFC where the voltage spectrum components of the demodulated signal are to be stored.

The output, SF, is normalized in such a way that its power will equal unity when the input signal is phase modulated by a single sinusoid with a modulation index of unity.

Error Message: None.

Note: This subroutine calls the DFT subroutine.

Purpose: This subroutine simulates a single-sideband (SSB) demodulator.

Fortran Calling Statement:

```
CALL DEMSSB (NFC,UFS,VF,FRD,SF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum,

VF = complex array of dimension NFC containing the voltage spectrum components of the demodulator input signal,

FRD = reference frequency of the demodulator,

and the output parameter is

SF = complex array of dimension NFC where the voltage spectrum components of the demodulated signal are to be stored.

Error Message: None.

Note: This subroutine calls the RANF function.

Purpose: This subroutine performs discrete Fourier transform between time and frequency domains.

Fortran Calling Statement:

```
      CALL DFT (N,X1,ISGN,X2)
```

where the input parameters are

N = dimension of the X1 and X2 arrays,

X1 = complex array of dimension N containing the input sequence to be transformed,

ISGN = sign in the exponent
= +1 for transform from frequency domain to time domain
= -1 for transform from time domain to frequency domain,

and the output parameter is

X2 = complex array of dimension N, where the transformed sequence is to be stored.

Restriction: N must be equal to 2**M, where M is an integer from 0 to 14 inclusive.

Error Message: If N violates the above restriction, an error message will be written on the standard output unit, and a normal exit will be taken from the subroutine.

Timing: Approximately 0.1 N milliseconds (on the CDC-6600 computer).

Note: This is a modified version of the discrete complex fast Fourier transform package programmed by L. David Lewis and Marie West of the Space Environment Laboratory, National Oceanic and Atmospheric Administration, U.S. Department of Commerce, Boulder, Colorado. Their subroutine package is a Fortran translation from an Algol procedure written by R. C. Singleton, "Algorithm 339, An Algol procedure for the fast Fourier transform with arbitrary factors," Commun. ACM, vol. 11, no. 11 (Nov. 1968), pp. 776-779, and vol. 12, no. 3 (March 1969), p. 189.

Purpose: This subroutine calculates frequency response of a Chebyshev-type bandpass filter. (cf. J. E. Storer, Passive Network Synthesis, McGraw-Hill, ch. 30)

Fortran Calling Statement:

```
CALL FRBPFC (NŌ,RPL,FC,BW,UFS,IFCMN,IFCMX,HF)
```

where the input parameters are

- NŌ = order of the bandpass filter (must be nonnegative but not greater than 100),
- RPL = ripple (dB) within the passband (must be nonnegative),
- FC = center frequency (must be positive),
- BW = 3-dB bandwidth (must be positive),
- UFS = unit frequency spacing (must be positive),
- IFCMN = minimum of frequency-component number for which frequency response is to be calculated,
- IFCMX = maximum of frequency-component number for which frequency response is to be calculated,

and the output parameter is

- HF = complex array of dimension IFCMX or greater where the calculated frequency response for UFS*(I-1) is to be stored as its Ith element.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Purpose: This subroutine simulates an amplitude-modulation (AM) modulator.

Fortran Calling Statement:

```
CALL MODAM (NFC,UFS,SF,FC,FLMI,VF)
```

where the input parameters are

- NFC = number of frequency components (must be 2**M, M = 2,3,...,14),
- UFS = unit frequency spacing of the spectrum (must be positive),
- SF = complex array of dimension NFC containing the voltage spectrum components of the modulating signal (must include at least one nonzero element),
- FC = center frequency of the modulated signal (must be smaller than (NFC-1)*UFS),
- FLMI = modulation index (must not exceed 1.0),

and the output parameter is

- VF = complex array of dimension NFC where the voltage spectrum components of the modulated signal are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the RANF function.

Purpose: This subroutine simulates a frequency-modulation (FM) modulator.

Fortran Calling Statement:

```
CALL MODFM (NFC,UFS,SF,FC,FDV,VF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum (must be positive),

SF = complex array of dimension NFC containing the voltage spectrum components of the modulating signal (must include at least one nonzero element),

FC = center frequency of the modulated signal (must be smaller than (NFC-1)*UFS),

FDV = frequency deviation,

and the output parameter is

VF = complex array of dimension NFC where the voltage spectrum components of the modulated signal are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the DFT subroutine and the RANF function.

Purpose: This subroutine simulates a frequency-shift-keying (FSK) modulator.

Fortran Calling Statement:

```
CALL MODFSK (NFC,UFS,NIB,IB,FC,FS,VF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum (must be positive),

NIB = number of information bits in a period (must not exceed NFC/2),

IB = integer array of dimension NIB containing the information bits,

FC = center frequency of the modulated signal (must be smaller than (NFC-1)*UFS),

FS = frequency shift (must be smaller than 2*FC and 2*(NFC*UFS-FC)),

and the output parameter is

VF = complex array of dimension NFC where the voltage spectrum components of the modulated signal are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the DFT subroutine and the RANF function.

Purpose: This subroutine simulates a phase-modulation modulator.

Fortran Calling Statement:

```
CALL MODPHM (NFC,UFS,SF,FC,FLMI,VF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum (must be positive),

SF = complex array of dimension NFC containing the voltage spectrum components of the modulating signal (must include at least one nonzero element),

FC = center frequency of the modulated signal (must be smaller than (NFC-1)*UFS),

FLMI = modulation index in radians,

and the output parameter is

VF = complex array of dimension NFC where the voltage spectrum components of the modulated signal are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the DFT subroutine and the RANF function.

Purpose: This subroutine simulates a single-sideband (SSB) modulator.

Fortran Calling Statement:

```
CALL MODSSB (NFC,UFS,SF,FR,VF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum (must be positive),

SF = complex array of dimension NFC containing the voltage spectrum components of the modulating signal (must include at least one nonzero element),

FR = reference frequency of the modulated signal (must be smaller than (NFC-1)*UFS),

and the output parameter is

VF = complex array of dimension NFC where the voltage spectrum components of the modulated signal are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the RANF function.

Purpose: This subroutine simulates a psophometer for commercial-telephone circuits. (cf. CCITT Recommendation, p. 53, 1973)

Fortran Calling Statement:

```
CALL PSPHCT (NFC,UFS,SF,PW)
```

where the input parameters are

NFC = number of frequency components,

UFS = unit frequency spacing of the spectrum in hertz,

SF = complex array of dimension NFC containing the voltage spectrum components of the input signal,

and the output parameter is

PW = psophometrically-weighted signal power.

Error Message: None.

Purpose: This subroutine simulates a psophometer for program-transmission circuits. (cf. CCITT Recommendation, p. 53, 1973)

Fortran Calling Statement:

```
CALL PSPHPT (NFC,UFS,SF,PW)
```

where the input parameters are

NFC = number of frequency components,

UFS = unit frequency spacing of the spectrum in hertz,

SF = complex array of dimension NFC containing the voltage spectrum components of the input signal,

and the output parameter is

PW = psophometrically-weighted signal power.

Error Message: None.

Purpose: This subroutine simulates a pulse-signal generator.

Fortran Calling Statement:

```
CALL SGPLS (NFC,UFS,FC,PRF,TR,TW,TF,DVF,VF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum (must be positive),

FC = center frequency of output signal (must be smaller than NFC*UFS),

PRF = pulse repetition frequency,

TR = pulse rise time,

TW = pulse width at the top,

TF = pulse fall time,

DVF = maximum frequency deviation (must be smaller than FC and NFC*UFS-FC),

and the output parameter is

VF = complex array of dimension NFC where the voltage spectrum components of the output signal are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the DFT subroutine and the RANF function.

Purpose: This subroutine simulates a white-Gaussian-noise generator.

Fortran Calling Statement:

```
CALL SGWGN (NFC,UFS,IFCMN,IFCMX,VF)
```

where the input parameters are

NFC = number of frequency components (must be 2**M, M = 2,3,...,14),

UFS = unit frequency spacing of the spectrum (must be positive),

IFCMN = minimum of frequency-component number for which components of VF are to be generated (must be 2 or greater),

IFCMX = maximum of frequency-component number for which components of VF are to be generated (must be IFCMN or greater, but not greater than NFC),

and the output parameter is

VF = complex array of dimension NFC where the voltage spectrum components of the output signal are to be stored.

Error Message: When this subroutine is called with improper input values, an error message will be written on a standard output unit, and a normal exit will be taken from this subroutine.

Note: This subroutine calls the RANF function.

GPO 912-678